1985

Multiobjective Routing Through Space And Time: The Mvp And Tdvp Problems

Carl Peter Keller

MULTIOBJECTIVE ROUTING THROUGH SPACE AND TIME:

THE MVP AND TDVP PROBLEMS

by

Carl Peter Keller

Department of Geography

Submitted in partial fulfilment

of the requirements for the degree of

Doctor of Philosophy

Faculty of Graduate Studies

The University of Western Ontario

London, Ontario

July 1985

ABSTRACT

This thesis concerns two routing problems, the 'Multiobjective Vending Problem' (abbreviated to MVP) and the 'Time Dependent Vending Problem' (abbreviated to TDVP).

To date, most research that deals with the problem of routing to cover a set of demand nodes has utilised a single objective approach, the objective being usually that of minimising distance or travel time. The set of nodes to be visited has been assumed known and specified. The MVP problem definition, based on a multiobjective solution approach, drops the latter assumption. The overall objective becomes that of identifying the trade-off relationship between two objectives, one to minimise some expression of route length, the other to maximise the coverage of nodes.

The study commences by discussing the advantages to utilising a multiobjective approach to optimisation research, stressing its potential role in spatial analysis. A number of general multiobjective research techniques are introduced. The MVP problem is defined mathematically, and a number of different solution approaches are discussed. Given present computing capabilities, solution by a heuristic based on the 'Constraint Method' is singled out as the most feasible approach to solve large MVP problems. Such a heuristic is designed, and is evaluated on a 25 node problem.

Problems of routing to cover a set of demand points have to date also predominantly focussed on problems where demand is uniform through time. The TDVP problem definition drops this assumption, allowing demand

potential at the different nodes to vary with time. Times of arrival at the nodes become an explicit consideration in the problem formulation, the objective being that of identifying the optimal route that maximises demand potential covered. A second heuristic is designed to solve this problem, and is again evaluated on a 25 node problem.

ACKNOWLEDGEMENTS

The completion of this thesis brings with it the end of a very enjoyable and memorable time at Western and in London. Thanks to all, faculty, staff and past and present grads at the Geography Department, fellow cavers and sailors as well as the extended 83 Centre Street family, for making this time what it was.

Especial thanks to Mike, Dick and Don not only for your endless support, advice and encouragement as mentors, but also for sharing many activities and time spent away from the Department.

Finally, thanks to Eileen who understood when the going got tough by showing continuing support and encouragement. It is to her that I would like to dedicate this thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

1.0 Multiobjective Routing Through Space and Time: The MVP and [illegible]VP Problems

1.1 Introduction

The planning of spatial phenomena and spatial systems, and the associated decision making processes, comprise a fundamental part of geographical research interests. Traditionally, geographers have pursued this research interest utilising a descriptive, qualitative and post-facto approach. It is only in the last two decades that some geographers have begun to place more emphasis on the quantification and optimisation of spatial planning and decision making processes (Abler et al., 1971; Harvey, 1973; Johnston, 1979). This shift in interest was made possible, and has been greatly assisted by the advent of high speed digital computing facilities in the 1960s. One topic of geographical investigation that has developed out of this more quantitative spatial enquiry deals with research into routing problems.

The general definition of a routing problem is discussed in some detail by Bodin et al. (1983: 79). Briefly summarised, a route is defined as a sequence of locations that are to be visited. At the most general level, there are no restrictions on when, or in what order the nodes must be arrived at. The problem is to construct a route or routes that minimise some cost, or maximise some feasibility.

The range of routing problems researched to date is broad. It includes, for example the Transportation Problem, the Shortest Path Problem, the Travelling Salesman Problem and Vehicle and Crew Routing and Scheduling Problems. Summaries of research undertaken can be found by Elmaghraby (1970), Turner et al. (1974), Golden and Magnanti (1977), Larson and Odoni (1981) and Bodin et al. (1983).

Geographers are not the only ones who have expressed an interest in routing problems. If anything, they are late arrivals to this field of enquiry, following operations researchers and engineers. Geographers do however appear to be placing a somewhat different emphasis on routing research. The interests of operations researchers tend to focus on the search for exact solution methods to narrowly defined problems, emphasising mathematical complexities and computational efficiencies. The geographers' interests stem more from an attempt to quantitatively analyse and evaluate real world spatial problems, classifiable as routing problems, that require planning and decision making. Geographers are therefore generally less interested in the mathematics and efficiency of solution techniques, but more interested in their real world applications.

A survey of existing routing research will show that the bulk of research undertaken is single objective in nature. The objective is usually that of minimising total distance or travel time. Real world routing problems are however often inherently multiobjective in nature (Current, 1981; Current et al., 1983). Why has a multiobjective approach to solving routing problems received so little research emphasis to date?

Steenbrink (1974) attributes the neglect of such a research approach to the fact that it is easier to make sufficient assumptions to narrow a problem down to one objective, than it is to define and set it up multiobjectively. This is true, but a multiobjective approach is often far more realistic. Shying away from attempting to define and solve routing problems multiobjectively because of the underlying complexities is not the only reason for a delay in multiobjective research. A more serious setback has for a long time been the lack of adequate computing facilities. It is therefore a combination of the advent and rapid development of high speed digital computers, and an increasing realisation of the ubiquitous nature and applicability of multiobjective problem solving, that has stimulated a recently growing interest in this field of enquiry (see Cochrane and Zeleny, 1973; Cohon, 1978).

The multiobjective nature of routing problems has been discussed by some. The general notion of decision analysis underlying multiobjective spatial phenomena is outlined by Nijkamp (1979). DeNeufville and Keeney (1973), Lee and Moore (1977), Aneja and Nairn (1979) and Jara-Diaz and Han (1980) discuss multiobjective approaches to evaluating transportation systems. Lee and Moore (1977) offer an indepth analysis of multicriteria school bussing models. Moore et al. (1978) introduce a transshipment problem which contains a number of conflicting objectives. Savas (1978) proposes that efficiency, effectiveness and equity are the three dominant criteria for evaluating public service performance, including public sector services with network and routing orientations. Current (1981) and Current et al. (1982, 1983) discuss and solve a number of multiobjective transportation network problems.

One of the problems discussed by Current (1981) that will serve as a useful illustration of the general nature of multiobjective problems is that of the Maximum Population/ Shortest Path (MPSP) problem. The MPSP problem has two objectives. The first is akin to that of the general shortest path problem, of finding the shortest possible route that connects two specified nodes in a network. The second objective is to maximise the number of nodes connected by the path, the path coverage. The optimal solution to the second objective alone is therefore to connect all nodes in a given network. It is assumed that the route has to start and terminate at a predetermined source and terminal node.

It is only in exceptional circumstances that the optimal solution to one of the objectives is also the optimal solution to the other. The objectives are therefore in conflict, and a number of optimal solutions will exist. Each optimal solution will represent a specific trade-off relationship between the two objectives.

It has been argued that it is notably geographers that are interested in more realistic solution approaches to applied routing problems. Still, the bulk of the multiobjective routing research, cited previously, has been undertaken by operations researchers and environmental engineers. Geographers, with the exception of a few, have so far not been actively involved in this area of investigation.

It is therefore one of the objectives of this thesis to outline the potential role of multiobjective research to spatial enquiry, and to discuss the role of geography in this form of analysis. Chapter Two introduces the advantages to utilising a multiobjective research approach. Chapter Three discusses and evaluates a number of different multiobjective research techniques. Throughout this Chapter, specific emphasis is placed on the applicability of the different multiobjective solution techniques to solving two multiobjective routing problems addressed specifically in this thesis, the MVP (Multiobjective Vending

Problem) and the TDVP (Time Dependent Vending Problem) problems. These two problems are introduced in detail below.

1.2 The MVP Problem

One of the classic routing problems that has in the past attracted considerable research interest, and continues to do so, is the travelling salesman problem (Gonzales, 1962; Bellmore et al., 1968; Held et al., 1971; Lin et al., 1973; Turner et al., 1974; Harvey et al., 1976; Bodin et al., 1983; Litke, 1984). The travelling salesman problem is conceptually defined as follows: Find the shortest complete circuit that connects a set of nodes so that every node is visited once and once only. The problem is defined mathematically in Chapter Four, and is discussed in more detail throughout the thesis.

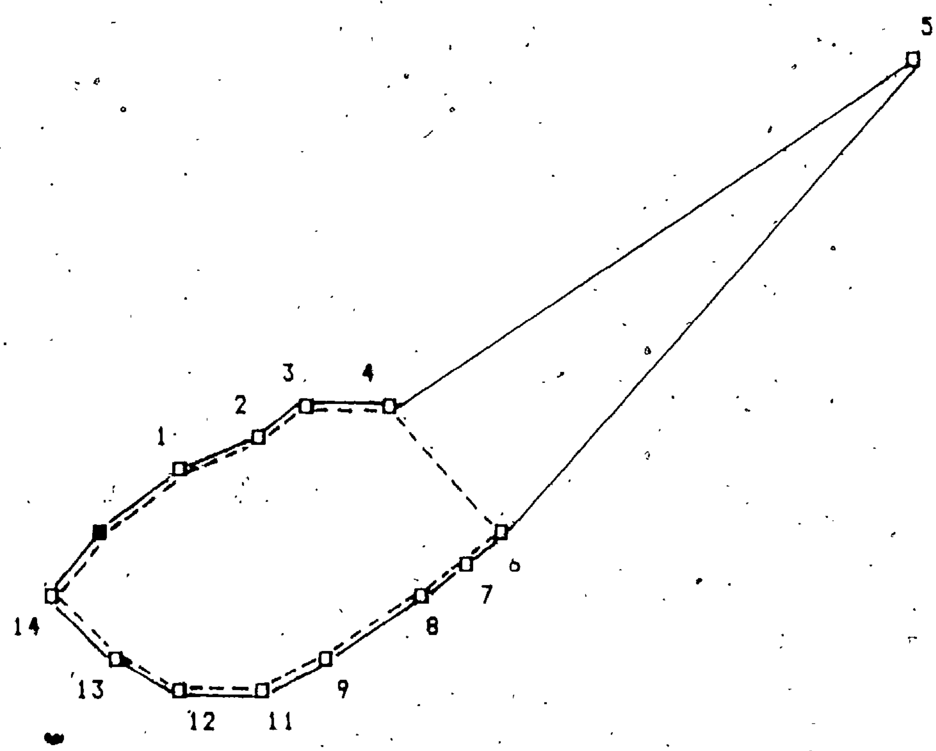
The applications of the travelling salesman problem in spatial analyses are broad and varied. At the most basic level, applications involve the identification of some single travelling salesman's most optimal path (Lin and Kernigham, 1972). More complicated analyses involve solving chromatic travelling salesman problems. They concern the partitioning of a set of nodes into subgroups, and the identification of each subgroup's travelling salesman circuit (Dantzig et al., 1959; Boyde, 1965; Hocking, 1972; Bellmore et al., 1974; Harvey et al., 1974; Golden, 1976). Other research combines travelling salesman analysis and location theory in an attempt at identifying one or a number of optimal depot sites (Clarke et al., 1964; Tillman et al., 1972; Gillett et al., 1974; Orloff, 1974).

An assumption underlying most of the above research is that the set of nodes to be visited is always given. The problems therefore reduce to finding one or a number of optimal travelling salesman routes that connect all of these nodes. This is not always realistic.

Consider the hypothetical example demonstrated in Figure 1.1. Shown is a 14 node problem and the optimum travelling salesman path. Let it be assumed that each of the 14 nodes has associated with it some reward potential, and that this reward potential can be collected upon arrival at that node. Figure 1.1 shows node 5 to be located at quite some distance from all the other nodes. The route penalty that must be accepted to travel towards and away from node 5 is therefore relatively large. The question may arise whether it is worthwhile travelling the extra distance to visit node 5, or whether this node should be excluded from the route sequence. The answer to this question will depend on the size of the reward offered at node 5, and on the trade-off relationship between reward and the penalty that must be accepted for travelling the links.

Problems that contain the above type of question clearly contain two objectives, and are therefore examples of multiobjective travelling salesman problems. The first objective is that of maximising reward to be collected by visiting as many nodes as possible, the second that of keeping the link penalty to a minimum. If the two objectives are measured in commensurable terms, say dollars, or if a trade off relationship is specified, then the problem can be solved as a single objective problem. Under these circumstances only one optimal solution will exist, and its identification will require finding that subset of nodes, and the associated travelling salesman path, that maximises the difference between the total accumulated reward and penalty.

Figure 1.1 A Hypothetical Example of a Travelling Salesman Route



□ = nodes

■ = the depot

—— = the travelling salesman route

* If the incentive to visit node 5 is small, then the dashed route may represent a preferred travelling salesman solution.

The two objectives are not always commensurable, and so the trade-off relationship between them is of central interest. One field where this problem arises as a practical concern is periodic marketing (Hay, 1971; Webber et al., 1973; Bromley et al., 1975; Ghosh, 1980, 1982). The problem is the following. A vendor searches for a travelling salesman circuit that connects an optimal subset of nodes so that the tour can be repeated at set intervals. The questions are therefore the following:- How much reward can the vendor expect to cover within a set time period, and how does reward potential change when the time interval is altered? This type of multiobjective definition is often far more realistic when analysing real world problems. The solution to this problem therefore comprises the second objective of this thesis.

This type of multiobjective vending problem (hereafter referred to as the MVP problem) is conceptually very similar to that of the Maximum Population/ Shortest Path problem researched by Current for his doctoral dissertation (Current, 1981). It differs from Current's work by the inclusion of an additional constraint, to ensure that the route presents a closed circuit. The MVP problem is defined mathematically in Chapter Four.

Current (1981) solves the MPSP problem utilising zero-one integer programming. This solution approach is discussed in some detail in Chapter Four. It is shown to be applicable only to problems which consider few nodes, and is somewhat inflexible. A number of alternative solution approaches are suggested. The discussion concludes that a heuristic represents a more feasible solution technique. This is argued to be especially true if MVP problems are to be solved that have to search amongst a large set of nodes. Chapter Five outlines such a heuristic. The objectives for the design were to make it user friendly by utilising an interactive programming approach, and to keep it computationally relatively simple. The advantages associated with interactive programming are discussed further on in this chapter. The heuristic was derived on the basis of trial and error. Chapter Six evaluates the heuristic on a 25 node problem.

The general definition of a routing problem given earlier argued that there should be no restrictions on when, or in what order nodes must be visited. Given no a priori restrictions on arrival, temporal considerations can be ignored in the problem formulation. This is not true if the time of arrival at a node becomes of primary importance, an issue addressed in the following section.

1.3 The TDVP Problem

Geographers have investigated locations and movements through space for a long time. An enquiry into locations and movements through time has been a more recent addition to geographical research, fostered by the 'Lund Approach' to time geography (Hagerstrand, 1970, 1973, 1975; Carlstein, 1974, 1975; Lenntorp, 1976) and an emerging interest in chronogeographical approaches to asking and solving geographical problems (Janelle, 1968, 1969, 1976; Abler et al., 1971; Relph, 1976; Tuan, 1977, 1978; Carlstein et al., 1978; Parkes and Thrift, 1980).

Part of this growing research interest involves the inclusion of time as an explicit consideration in routing problems, most notably to date in 'routing and scheduling' problems (Bodin et al., 1983). This class of problems deals with analyses where the sequence of nodes to be visited may be constrained by a precedence relationship, or where individual nodes have to be visited within a certain time window, or at some definite time (Wren and Holliday, 1972; Orloff, 1976; Baker, 1980; Assad et al., 1981; Christofides, 1981; Swersey and Ballard, 1982; Bodin et al., 1983).

An example of a routing problem with underlying precedence relationships is that of dial-a-ride. Here, the precedence relationship is that passengers must be collected before being dropped of. A routing problem that contains time windows is one where the nodes must be visited during, before, or after some specified time. An example of a definite service time routing problem is that of transit routing, where different stops are to be arrived at and departed from at specified times.

There exists a class of time dependent routing problems so far not researched. Consider the problem of identifying the optimal travelling salesman route for a mobile vending unit. The objectives underlying the planning problem are twofold. The first is to maximise the exposure to potential customers, the second to keep the 'dead time' of the vending operation, the time spent travelling the links, to a minimum. The time component enters the problem two ways. First, the operation is limited by a maximum operating time. Second, demand (the number of customers potentially interested in the vendor's service) is dependent on the time of day. In other words, demand is not uniform through time. Given this underlying demand cycle, the time of arrival at the individual nodes becomes an explicit consideration in the problem formulation. The objective is therefore to identify the subset of nodes that will maximise demand, and the route that connects them, while remaining within the defined operating time. Intuitively, the optimum solution would be one where as much time as possible is spent vending during the peak(s) of the demand cycle, and where most of the necessary time spent travelling the links occurs at low demand.

A second example involves the timing of a political canvassing trip prior to an election. A candidate will try and maximise his/her exposure to the electorate within the pre-election campaign period. The probability of people turning up to listen to the candidate is dependent on the time of day, and the day of the week. Furthermore, the timing and location of the release of election promises and policy statements may be crucial. The objective is to identify the set of places to visit and the route that will maximise the candidate's exposure to the electorate within the campaign period, and that will allow the candidate to be at the right place at the right time to release crucial election statements.

Numerous other examples of real world routing problems that are dependent on this type of time consideration exist. Most of these are in sales and marketing, some are as diverse as the planning of a touring holiday. A tour operator may have the choice of visiting n different places when planning any one tour. It is only possible to visit a subset of all possible places given constraints of budget and time. The nodes, for example cities, will always contain some attractions and opportunities (the general image, historical buildings, museums...). Others may be more time specific (Munich Beerfest, Carnival in Rio de Janeiro, the performance of a concert...). The opportunity potential when visiting any one city therefore depends on the time of arrival, and the tour operator's decision process will have to include an evaluation of the relative importance of the different attractions and events offered at the different nodes at different times. His objectives will be to maximise opportunity while minimising cost.

The research problems cited above cannot be classified as scheduling problems. This is because, unlike scheduling problems, they still assume that any node in the problem can be visited at any time. No time windows or precedence relationships are imposed. Time does however become an explicit consideration since the incentive to visit a node is dependent on the time of arrival at that node. Problems of this type have so far not received any research attention, but offer an interesting and realistic research approach. A discussion of this problem, the time dependent vending problem (hereafter referred to as the TDVP problem) is therefore the third objective of this thesis.

The TDVP problem is defined mathematically in Chapter Four. To keep the definition simple and general, a number of assumptions have been made. It is assumed that the same time dependent function applies to the entire problem space. This implies that demand or reward potential increases or decreases simultaneously at all nodes through time. It is further assumed that the reward potential at any one node depends only on the time of arrival, and does, therefore, not fluctuate throughout the stay. Possible solution approaches are discussed in Chapters Three and Four. A heuristic utilising interactive programming is again suggested to be the most appropriate and feasible solution approach, especially given that a large set of possible nodes may have to be evaluated. The design of such a heuristic is discussed in Chapter Seven. The TDVP heuristic is evaluated for the same 25 node problem used when evaluating the MVP heuristic.

Both the MVP and TQVP heuristics were designed to be interactive. The advantages to such a programming approach are outlined in the following section..

1.4 Interactive Programming

Most optimisation algorithms written in the past have been developed for use on computers in a batch type environment. This implies a 'one-shot approach'. Given the specification of an objective function, constraints and a data set, the submission of the program will result in the output of one optimal answer. Any form of sensitivity analysis or problem modification requires a new specification, and a re-run of the entire program. Batch-type optimisation programs have proven successful, but usually require considerable understanding of the computer. The dialogue with the user is cumbersome and unfriendly, and the programs allow little flexibility. This is acceptable if the dominant research interests involve mathematical and computational considerations. It is less acceptable if the algorithms are to be used on a frequent basis to assist in real world planning and decision making. A different programming approach is therefore required,

A programming approach that appears to be increasingly more fashionable is that of man-machine interaction, including interactive graphics. This man-machine interactive approach involves direct dialogue between the user and the computer in the form of verbal and graphic response questions. It permits instantaneous visualisation of data and computational results. The user is able to interact with the program by inspecting and evaluating solutions derived, and by subsequently testing for changes in that solution when modifying one or a number of parameters.

Use of interactive graphics and a man-machine interactive programming approach in transportation research and network analysis has already been described by Krolak *et al.* (1972), Schneider (1974), Kornhauser and Hess (1978), Rapp *et al.* (1978) and Babib *et al.* (1982). Such an approach offers a considerable improvement over more conventional batch-type programming, and represents a means by which geographers can increase the flexibility and adaptability of spatial optimisation theory. The MVP and TDVP heuristics represent examples of such interactive programming.

CHAPTER TWO

2.0 Multiobjective Research and Spatial Enquiry

2.1 Introduction

This chapter commences by outlining the conceptual and mathematical differences between single objective and multiple objective approaches to solving optimisation problems. A number of advantages are noted when utilising a multiobjective approach. The growth and development through time of research concerning multiobjective analysis are discussed. The present and future potential roles of multiobjective research techniques in geographical enquiries are noted.

2.2 The Conceptual and Mathematical Differences Between Single Objective and Multiple Objective Research Approaches

The conceptual difference between single objective and multiple objective approaches to optimisation analysis can best be demonstrated by reference to an example. In the fire station location problem researched by Schilling (1976), the overall objective was to locate a site or sites for one or a number of new fire stations, using two goals. One goal is that of maximising the coverage of real estate, the other that of maximising the coverage of population. Real estate value tends to peak in the industrial subdivisions of a city and in the central business district, while the bulk of the population tends to be concentrated in the residential suburbs for the largest part of the day. The two dominating goals are therefore in conflict or in competition.

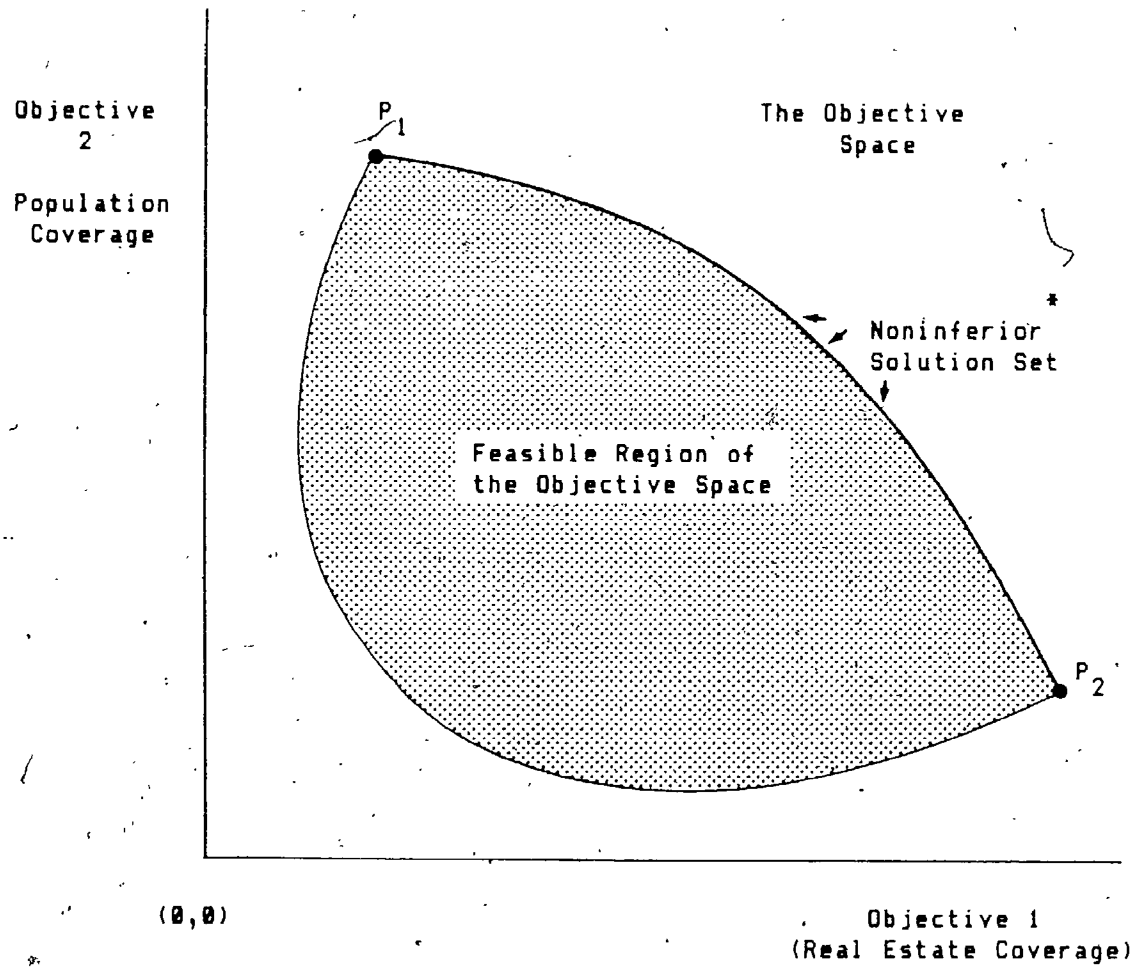
It is possible to evaluate this type of problem as a single objective problem by stating the cost of a life relative to the cost of real estate, that is by making the two goals commensurable. This will allow the two goals to be judged against each other. They can therefore be combined into one single objective, that of maximising overall coverage for the relationship specified. The result of such an optimisation procedure will yield one optimal answer.

The drawback to utilising a single objective approach to solve problems such as the one outlined above is that the two dominant goals within the overall objective function are rarely commensurable. Whereas real estate can readily be expressed in, for example, monetary terms, the life of a human being cannot. The two goals are therefore non-commensurable, and some arbitrary value judgement is required to compare the cost of a life to that of real estate. Very few people are willing to make such a judgement.

The idea underlying a multiobjective approach is not to make such a value judgement by specifying the relationship between the two goals, but to evaluate the trade off between covering more life against covering more real estate. The idea is to evaluate for a number of objectives ranging from the extreme of covering only for real estate to the other extreme of covering only for population. Utilising such an approach, there no longer exists simply one single optimal solution. Instead a solution set of best-compromise answers is identified.

Figure 2.1 depicts such a solution set for a two-goal problem that can yield real-value answers. Shown is an area within the objective space that comprises all possible feasible solutions to the research problem. A subset of this feasible region, the darker line in Figure 2.1, can be identified as containing the set of best-compromise solutions. A discrete version of such a two-goal problem would imply

Figure 2.1 A Multiobjective Solution Space and the Noninferior Solution Set



: the area inside this hull contains the set of all feasible solutions

P_1 : optimal solution when covering for population only

P_2 : optimal solution when covering for real estate only

* : this set is also known as the Trade-Off Curve, the Set of Best-Compromise Solutions.

that only a finite set of points within the feasible region pose as possible solutions.

This set of best-compromise solutions is based on the concept of noninferiority, which can be explained by the following definition:

> A feasible solution to a multiobjective programming problem is noninferior if there exists no other feasible solution that will yield an improvement in one objective (goal) without causing a degradation in at least one other objective (goal), (Cohon 1978:70).

Cohon (1978: 69) shows that the idea of noninferiority is very similar to the concept of dominance, or to efficiency in statistics and economics. The set of best-compromise solutions is often referred to as the noninferior solution set, that set of points which is not dominated by any other solution in the feasible region of the objective space. Different points on the noninferior solution set show how much of one objective must be traded off or sacrificed to obtain a specified gain in another objective. If one were to treat the two goals or objectives as commodities, such as apples or oranges, then the noninferior solution set is nothing more than a trade-off curve or a production frontier in economic terms.

To conclude, the important conceptual difference between single objective and multiple objective approaches to solving optimisation problems is that, whereas one optimal answer can be identified in the first case, the latter offers a set of optimal answers which yield an understanding of the trade off relationships.

Mathematically, the difference between the two approaches can be demonstrated by looking at the differences in their overall objective functions. The objective function, Z, of a single objective optimisation problem with n decision variables x_i, $(i=1,\dots,n)$, can be expressed in the following general form:

$$\text{Optimise } Z(x_1,x_2,x_3,\dots,x_n)$$

The general expression of the overall objective function, Z, of a multiobjective optimisation problem with n decision variables and p independent goals or objectives, on the other hand, is defined as

$$\begin{aligned}\text{Optimise } Z(x_1,x_2,x_3,\dots,x_n) = [&Z_1(x_1,x_2,x_3,\dots,x_n),\\ &Z_2(x_1,x_2,x_3,\dots,x_n),\\ &\dots\dots\dots\dots\dots\dots,\\ &Z_p(x_1,x_2,x_3,\dots,x_n)]\end{aligned}$$

The overall multiobjective function therefore consists of a combination of a number of separate single objective functions. The important thing to note is that these individual objective functions are merely listed; since the relative importances of the different objectives are not specified, they are not added, multiplied or combined in any way.

2.3 Advantages to Utilising a Multiobjective Research Approach*

To discuss the advantages underlying a multiobjective approach to solving optimisation problems, let us examine optimisation analysis in the planning context, that is, in the context of a problem that requires a decision making process. Let it be assumed for simplicity's sake that two dominant actors, the analyst and the decision maker are involved in

* The discussion of this section is based, in part, on chapters I and II of Cohon (1978)'s book on multiobjective programming (Cohon, 1978).

a decision making process. The analyst's job is to develop and run the tools required for the actual analysis of a problem. The decision maker specifies the problem, and ultimately decides on the preferred course of action.

To date, the bulk of tools available to the analyst have been single objective in nature. This implies that all objectives, criteria or attributes that are to be considered in the analysis must be commensurable, that is, they must be evaluated in terms of a single unit of measurement. However, it was noted above that there exist planning situations where this assumption of commensurability does not fit.

Let us look at a second problem, that of locating a site on which to develop an industry that will involve some environmental degradation. The costs of site purchase, construction considerations, locational efficiency, etc. associated with any one specific site can easily be quantified in monetary terms. It is however quite difficult, if not impossible, to evaluate the environmental impacts in monetary terms. The two objectives, monetary considerations and environmental degradation are, therefore, non-commensurable. If this sort of a problem was to be solved by a single objective approach, the analyst, often with little help from the decision maker, must place a monetary value on environmental impact. He will return to the decision maker one optimal solution. Should this solution encounter strong opposition from the public, then the decision maker can simply blame the analyst's value judgement, and the latter will be the scapegoat of the critique.

If the analyst had adopted a multiobjective approach, he would return to the decision maker a number of different solutions in the form of a trade-off curve. It is now the decision maker, who will presumably have a better knowledge of the politics underlying the problem, who will have to evaluate amongst the different alternatives and make a decision. Should the decision now backfire, it is the decision maker who will have to take the blame.

Basically therefore, four advantages to utilising a multiobjective approach to solving optimisation problems can be identified. First, as noted, such an approach allows for the evaluation of problems that contain non-commensurable objectives. Second, a multiobjective approach places the onus of preference stating between conflicting or competing objectives on the decision maker, and away from the analyst. Third, a multiobjective approach gives a better insight into a problem by including sensitivity analysis, which allows the decision maker, even if specifying a value judgement between non-commensurable objectives, to look at other solutions that are close to but not quite optimal for that preference. Fourth, Cohon (1978) notes that the most powerful advantage underlying a multiobjective approach to analysis is that real-world problems are multiobjective in nature and should be evaluated as such.

2.4 Multiobjective Research and Geographical Enquiry

Initial research interests and the general growth of work in multiobjective analysis can be traced by a close inspection of a bibliography on 'Multiple Criteria Decision Making' compiled by Zeleny (1975a). This bibliography includes all the work published on multiple criteria decision making prior to the middle of 1975, including research notes and working memoranda appearing in regularly maintained institutional series, but excluding literature on multidimensional scaling and multiattribute models of consumer behaviour. Table 2.1, compiled from this bibliography, shows the growth of the above-mentioned

Table 2.1 Output of Publications on Multiobjective Research until 1975

Year of Publication	Number of Publications	Cumulative Number of Publications
Pre 1955	5	5
1956-1960	10	15
1961-1965	26	41
1966-1970	97	138
1971	48	186
1972	46	232
1973	112	344
1974	74	418
1975 (first half)	72	490

Compiled from Zeleny (1976) "Multiple Criteria Decision Making Bibliography- 1975" research through time. The table shows that some of the earliest work in multiobjective research was undertaken in the 1950s, most notably by Kuhn and Tucker (1951), who discuss the conditions underlying noninferiority, and by Koopmans (1951) who assesses activity analysis of production and allocation. The table shows an initial slow growth of research interest until the late sixties, but a considerable output of research by the mid seventies, the time when the bibliography was compiled.

The first conference on multiple criteria decision making was held in Kyoto, Japan, organised to demonstrate the validity and importance of the subject (Zeleny, 1976: Chpt. III). Four conferences on multiple criteria decision making have subsequently been held, and research concerning multiobjective problems continues to be published in leading journals. This includes special issues completely devoted to the topic, such as in the journal Management Science (1977) and the journal Computers and Operations Research (1980).

The proceedings of the above mentioned conferences (see Zeleny, 1976; Zionts, 1978; Fandel and Gal, 1980; Morse, 1981 and Hansen, 1983), as well as the literature on multiobjective research at large, show that the bulk of research to date has been undertaken in the disciplines of operations research or management sciences, engineering, economics, and psychology. Research in operations research and engineering tends to be mathematical and theoretical in nature, and of late has focused specifically on attempting to solve the multiobjective simplex algorithm. Research in economics tends to focus on problems of resource allocation and public investment problems. Psychological research concerns itself predominantly with scaling methods, most notably the multidimensional scaling of attributes.

Perhaps ironically, while problems of a geographical nature have received some attention in multiobjective research, this research was not undertaken by geographers, but rather by engineers such as Haimes et al. (1974; 1975) and by regional scientists and economists such as Nijkamp (1977; 1978; 1979), Nijkamp et al. (1976; 1977; 1979) and Rietveld (1980). One exception is the Department of Geography and Environmental Engineering at the Johns Hopkins University, Baltimore, where considerable research on multiobjective analysis has been carried out (see for example Cohon et al., 1975; Schilling, 1976; Cohon, 1978; Schilling et al., 1980; Current, 1981 and Current et al., 1982; 1983).

The bulk of multiobjective research in spatial analysis concerns the application of goal programming (see for example Courtney et al., 1972; McGrew, 1975; Schuler et al., 1975; 1977; Barber, 1977; Dane et al. 1977 and Leinbach et al. 1983). As will be noted in Chapter Three, goal programming is however not a multiobjective technique in the strict sense. With the exception of research at the Department of Geography and Environmental Engineering at Johns Hopkins, and some work by Church and Huber (1980), few geographers have therefore published work undertaken on the development of multiobjective geographical tools.

2.5 Conclusion

This chapter has outlined the conceptual and mathematical differences between single objective and multiobjective approaches to optimisation analysis, and to solving decision making problems. The advantages underlying a multiobjective research approach were stated.

It was noted that, while multiobjective spatial problems have attracted the attention of scientists working in multiobjective analysis, geographers themselves have to date not shown much interest in this sort of enquiry. It is suggested that a focus of research efforts by geographers towards the development of tools for multiobjective spatial analysis represents a possible avenue for future research in theoretical and applied geography.

CHAPTER THREE

3.0 Multiobjective Research Methods

3.1 Introduction

The objective of this chapter is to discuss a number of different approaches to solving multiobjective research problems. Two different types of general solution approaches are distinguished on the basis of information flows, and the types of solutions expected from the analysis. Examples of specific techniques for each of the two general solution approaches are discussed. The different methods are evaluated in terms of their applicability to solving the research problems of central interest to this thesis, the MVP and the TDVP problems. The chapter concludes by identifying the solution approach thought most appropriate to solving these two problems.

3.2 Two General Solution Approaches

Two general solution approaches can be distinguished on the basis of information flows, and the type and nature of the results expected from a multiobjective optimisation analysis.* The two general solution approaches are not mutually exclusive, and there exists no sharp distinction between them.

To understand the difference between the two approaches, let us once again distinguish between the analyst and the decision maker as two actors in a decision making process. A multiobjective approach to solving an optimisation problem has been agreed upon. Two situations may

---

*: For a more detailed discussion of this topic see Cohon, 1978.

now arise. First, the decision maker may suggest some value judgement on all, or a number of the attributes or objectives in the multiobjective research problem. The analyst will then find the best-compromise solution for the preferences articulated by identifying that point on the noninferior solution set that satisfies the specified preferences best. Subsequent sensitivity analysis will identify marginal alternatives, and will allow for a better insight into the problem. This type of approach assumes that the decision maker has a reasonable knowledge of, and some set ideas about the optimisation problem at issue. The information flow is basically from the decision maker to the analyst. Problems of this nature are generally referred to as 'Preference-Oriented Problems'.

Alternatively, the decision maker may have little insight into the optimisation problem at stake. The analyst is, therefore, simply given the different objectives, and is asked to generate the entire noninferior solution set. Given the entire solution set, the decision maker will be in a position to understand thoroughly the relationships between the different objectives, or the relative trade offs. In this case the information flow is argued to be from the analyst to the decision maker (Cohon, 1978). Techniques that solve this type of a problem are generally referred to as 'Noninferior Solution Set Generating Techniques'.

3.3 Preference-Oriented Techniques

As noted above, preference-oriented techniques identify the best-compromise solution for some explicit preference statement or value judgement between objectives, and subsequently perform a sensitivity analysis.

Prior to discussing techniques which solve this type of problem, it helps to first familiarise oneself with the concept of multiattribute utility functions. A detailed discussion of the general concept of multiattribute theory is covered in a book by Keeny and Raiffa (1976), and in a review by Farquhar (1977). A brief discussion is given below.

A simple utility function is basically a mathematical function that states the relative preference of one alternative over another; one may be more preferred or less preferred, or a person may be indifferent. A multiattribute utility function is then

> "a mathematical statement that indicates the utility of all combinations of values for the various attributes or objectives that are under consideration, where there are many attributes or objectives associated with an alternative", (Cohon 1978:167).

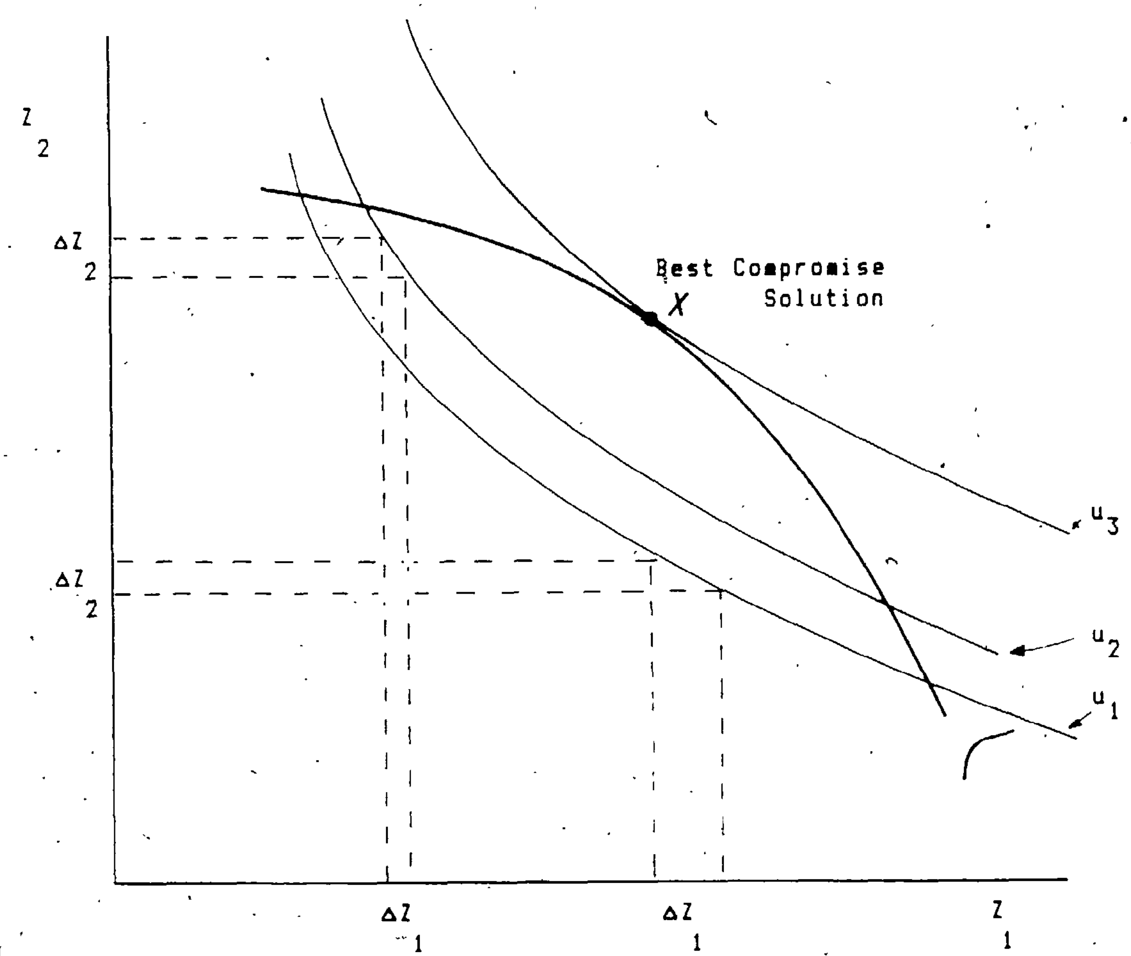
A multiobjective utility function can best be thought of as an isopreference curve or an indifference curve as shown in Figure 3.1. Shown are three indifference curves for three levels of utility, u1, u2 and u3 for the two objectives Z1 and Z2. The three utilities have the following relationships:

$$u1 < u2 < u3$$

$$\text{and } u2 - u1 = u3 - u2$$

The utility functions are concave and utility increases in a northeasterly direction. The indifference curves show a number of things. First, as one moves away from the origin, one has to move relatively further in order to gain another unit in utility; that is u2 - u1 = u3 - u2 but the gap between u2 and u3 is larger than the gap between u2 and u1. Second, the indifference curves tell about the

Figure 3.1 Multiobjective Utility Functions



marginal rate of substitution. It shows how much of one objective has to be sacrificed in order to obtain a specified gain in another objective by moving along the isopreference curve (see ΔZ1 compared to ΔZ2 in Figure 3.1). Third, and most important, the indifference curve allows us to identify the best-compromise solution for the stated preference between objective one and objective two.

This best-compromise solution is found by the 'Northeast Rule'. This rule states that the point of maximum utility for a given preference statement, or for given constraints on all but one objective, is that point within the feasible region of the objective space which has no other feasible point to the north or east of itself, and which also maximises utility. The best compromise solution can be found graphically by successively increasing overall utility by pushing the indifference curve towards the northeastern quadrat of the objective space until it is tangent to the noninferior solution set (see point X in Figure 3.1). Given that a graphical solution to the problem is not the preferred course of action, the question arises as to what programming technique should be adopted to find this point.

A number of such programming techniques exist. The method of 'Goal Programming' and the 'Surrogate Worth Trade Off Method' are the two that are most likely to be considered.

3.3.1 Goal Programming

Goal programming is one of the most well known multiobjective research methods, and the one most frequently utilised by geographers. A full discussion of the technique can be found in Charnes and Cooper (1961), Johnson (1968), Lee (1972) and Kornbluth (1973). Examples of applications are by Courtney et al. (1972), McGrew (1975), Barber

(1977), Lee and Moore (1977), Schuler et al. (1977), Kwak and Schniederjans (1979) and Leinbach et al. (1983).

Goal programming is generally utilised where there are a number of competing goals or objectives, as for example in the case of different land use development possibilities, where the overall aim is that of meeting all the goals to the greatest extent possible. It is thus a method useful in multiobjective analysis where the overall objective is to choose the most desirable plan from a set of possible options, each with its own characteristics in terms of a number of contrasting criteria.

Traditionally this type of a problem has been solved by the method of cost-benefit analysis. This method involves the quantification of all relevant parameters in commensurable and usually fiscal terms. It is notably Cohon and Marks (1975), Nijkamp (1975), and van Delft and Nijkamp (1976, 1977) who have proposed the alternative goal programming approach.

Nijkamp (1975) proposes commencing by normalising all criteria that the problem is concerned with, and subsequently weighting the relative importance of each in terms of the overall plan. Using the two matrices of normalised variables and normalised weights, two further matrices, the concordance matrix and the discordance matrix, are constructed. Logical inspection of the concordance and discordance matrices will yield the ultimately most preferred plan. The process does involve some arbitrary decisions, such as the relative weighting of the different criteria (see van Delft and Nijkamp 1976, 1977). The solution sensitivity must therefore be evaluated carefully by solving the problem at stake with different weightings. If the solution sensitivity is not evaluated, a specified set of goals can easily lead to an inferior solution, and the optimum would thus never be reached.

Cohon notes that goal programming is not a solution method in itself; it rather represents a problem solving approach which assesses competing or conflicting objectives involving contrasting entities in an attempt at solving all of the objectives to the greatest extent possible simultaneously. Strictly argued, goal programming is therefore in actual fact not a multiobjective programming technique, although generally perceived as such (Cohon, 1978: 187). The method is not readily applicable to the two problems of central interest to this thesis, those of optimising two or more objectives simultaneously that are dependent on each other. Dependence arises in the MVP and TDVP problems since a change in one objective implies a change in the other objective, that is, for every change in overall reward one must accept a change in penalty. The method of goal programming is therefore rejected as a technique for solving the MVP or TDVP problems.

3.3.2 The Surrogate Worth Trade Off Method

The surrogate worth trade off method was developed by Haimes and Hall (1974) and Haimes <u>et al.</u> (1975). Instead of asking the decision maker to specify goals for the different objectives, this method gets the decision maker to articulate a range of preferences. The analyst then evaluates for that part of the noninferior solution set which covers the range of choices articulated. Subsequently, the decision maker reacts to this subset of the overall noninferior solution set, and identifies the most preferred course of action within the specified preference range.

This method requires the evaluation or estimation of part of the noninferior solution set and is, therefore, merely a special case of the noninferior solution set generating technique, a topic which will be covered further on in this chapter. The method will therefore not be discussed further here. Examples of the applicability of the surrogate worth trade off method can be found in Haimes (1977) and Cohon (1978).

3.3.3. Discussion

The sensitivity analysis underlying preference-oriented techniques often results in these problems being referred to as iterative preference oriented techniques. These procedures adopt some formal mechanism by which the decision maker interacts with the analyst in an iterative fashion. The objective is to move from one noninferior solution point to another in a direction specified by the decision maker, until he is satisfied that he has reached the best-compromise solution for the problem to be evaluated. These techniques comprise one of the most active research areas in multiobjective analysis (Cohon, 1978: 200), and several techniques other than the two discussed above have been developed. Examples include Briskin's Method (Briskin, 1966), Geoffrion's Method (Geoffrion, 1967), the 'Step Method' (Benayoun <u>et al.</u>, 1971), 'Interactive Goal Programming' (Dyer, 1972), 'Sequential Multiobjective Programming Systems' (Monarchi <u>et al.</u>, 1973), Belenson and Kapur's Method (Belenson and Kapur, 1973) and the approach outlined in Zionts and Wallenius (1976). All these methods fall somewhere in between the concept of goal programming and the surrogate worth trade off method. Goal programming has however been rejected as a possible solution approach to the MVP or TDVP problems, and the surrogate worth trade off method was noted to be nothing more than a special case of a noninferior solution set generating technique. Preference-oriented techniques are therefore not discussed any further in this study since, as shown below, noninferior solution set generating techniques offer a more appropriate method for solving MVP and TDVP problems.

3.4. Noninferior Solution Set Generating Techniques

In principle, calculation of the entire noninferior solution set requires the evaluation of an optimisation problem for every point on that trade-off curve. In a problem that deals with real values, this would make the noninferior solution set infinite. In an integer problem, the set is finite, albeit often extremly large. Given this potential size of a solution space, it is, therefore, generally accepted as satisfactory to generate an approximation of the noninferor solution set by evaluating a subset of all solution possibilities (Cohon, 1978).

Four generally accepted methods exist for such an estimation of the noninferior solution set. Three depend on the conversion of the multiobjective problem into a number of single objective problems. They are the 'Weighting Method', the 'Constraint Method' and the 'NISE Method'. A fourth method, the 'Multiobjective Simplex Method', operates directly on the multiobjective problem.

The multiobjective simplex method, based on research by Philip (1972) and Ecker and Kouada (1975), is an active research area amongst mathematical programmers at present. A number of simplex-based algorithms have been presented for example by Holl (1973), Evans and Steuer (1973), Zeleny (1974) and Yu and Zeleny (1975). However, as Cohon notes, "the development of such techniques is a complex and intriguing mathematical problem that is not yet entirely solved" (Cohon, 1978:

140). This is still true seven years later. Solving the noninferior solution set via a simplex-based algorithm is, therefore, not considered to be a realistic solution possibility for the purpose of this study, and this solution technique is not discussed further.

The following three sections comprise a brief introduction to the 'Weighting Method', the 'Constraint Method' and the 'NISE Method' respectively.

3.4.1. The Weighting Method

One method of evaluating the noninferior solution set, first recommended by Zadeh (1963), and used for example by Marglin (1967) and Major (1969), is the method of relative weighting of the different objective functions.

To describe how the weighting method works, let us return to Schilling's fire station location problem, discussed in the previous chapter. It was noted that this problem could be expressed as a single objective problem if a judgement was made concerning the cost of life relative to that of real estate. The problem can then be expressed as follows:

$$\text{Maximise} \quad Z(W) = Z_1 + W*Z_2 \qquad 3.1$$

$Z(W)$ stands for the optimal solution given that W units of Z_1 are worth 1 unit of Z_2.

It is possible to turn this single objective expression into that of a straight line by the following simple transformation.

$$Z_2 = Z(W)/W - (1/W)Z_1 \qquad 3.2$$

This straight line can be interpreted as an indifference line with slope -1/W and an intercept at Z(W)/W. A close inspection of Figure 3.2 shows that it is possible to push this indifference line in a northeasterly direction until it is tangential to the feasible region of the objective space based on the northeast rule. The point of tangency, point C, is equivalent to the point of maximum utility, or the best compromise solution for weight W.

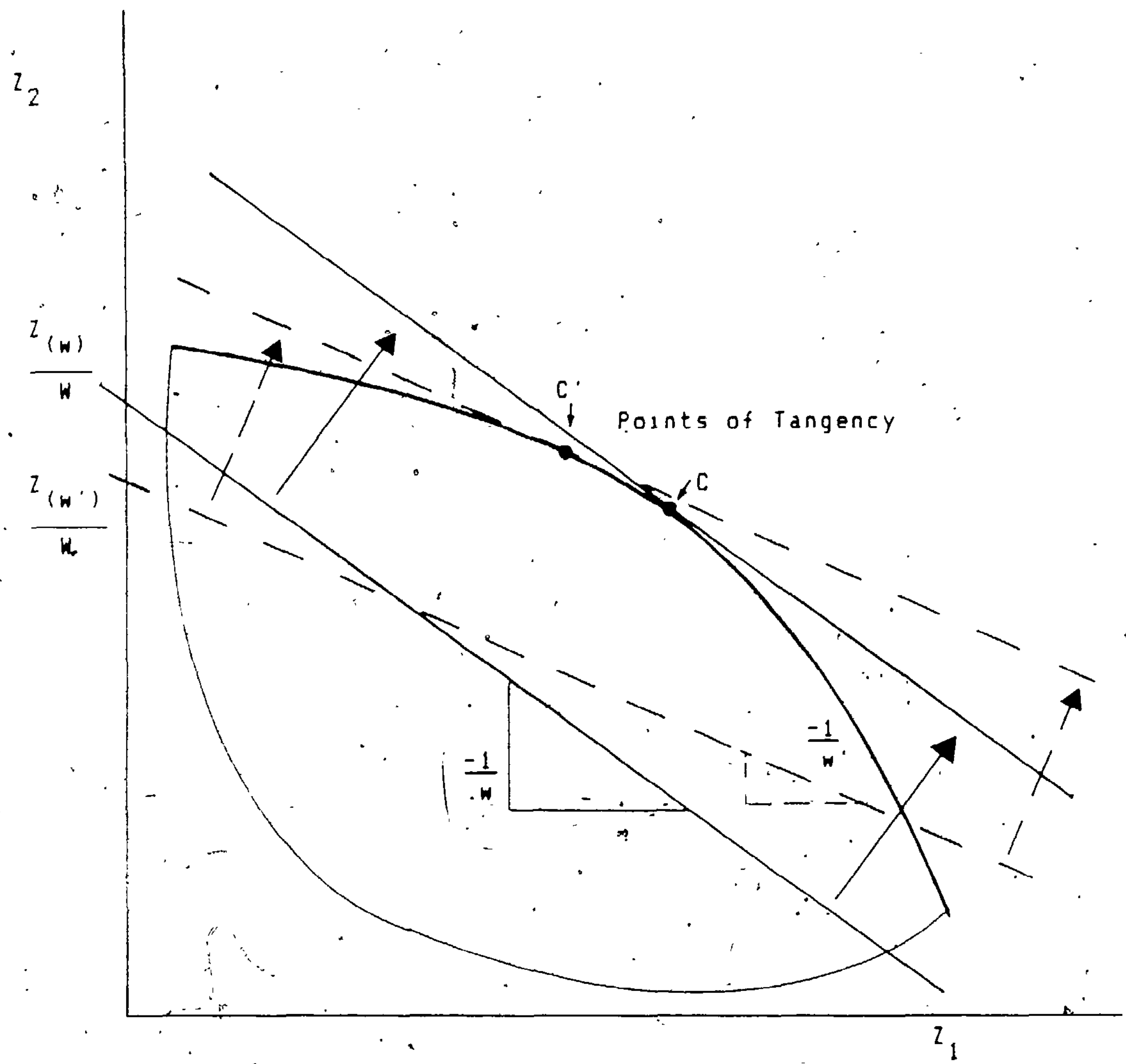
Should the value of a life relative to that of real estate be changed from W to W', then the slope of the indifference line would change. Pushing this new indifference line northeast, it will be tangential to the feasible region of the objective space at another point, point C' (see Figure 3.2). It is possible by this method to evaluate a number of different best-compromise solutions, or different points of the noninferior solution set, by continually changing the value of W.

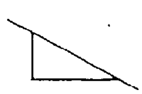
The actual weighting method operates on a slightly modified version of the technique described above. The relative objectives are weighted by assigning positive weights whose sum equals unity to the two objective functions. Thus

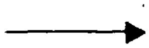
$$Z(W) = W*Z_1 - (1-W)*Z_2 \tag{3.3}$$

Given the above single objective expression, it is very easy to solve for the endpoints of the noninferior solution set by simply assigning W first the weight of unity, and subsequently the weight of zero, the equivalent of optimising each of the two objective functions individually. Given the endpoints thus derived, and given that the range of W lies between zero and unity, it is now possible to approximate the

Figure 3.2 Generating a Noninferior Solution Set Graphically by the Weighting Method



 = Slope

 = the indifference line is pushed <u>Northeast</u> until it is tangent to the noninferior solution set

noninferior solution set in between the two endpoints by incrementing W in a stepwise manner from zero to unity at intervals of 1/k, where k+1 stands for the number of points that will approximate the set.

Mathematically, for each weight W, the point of maximum utility or the best-compromise solution can be derived by some optimisation procedure. The exact nature of this procedure will be discussed in the following two chapters.

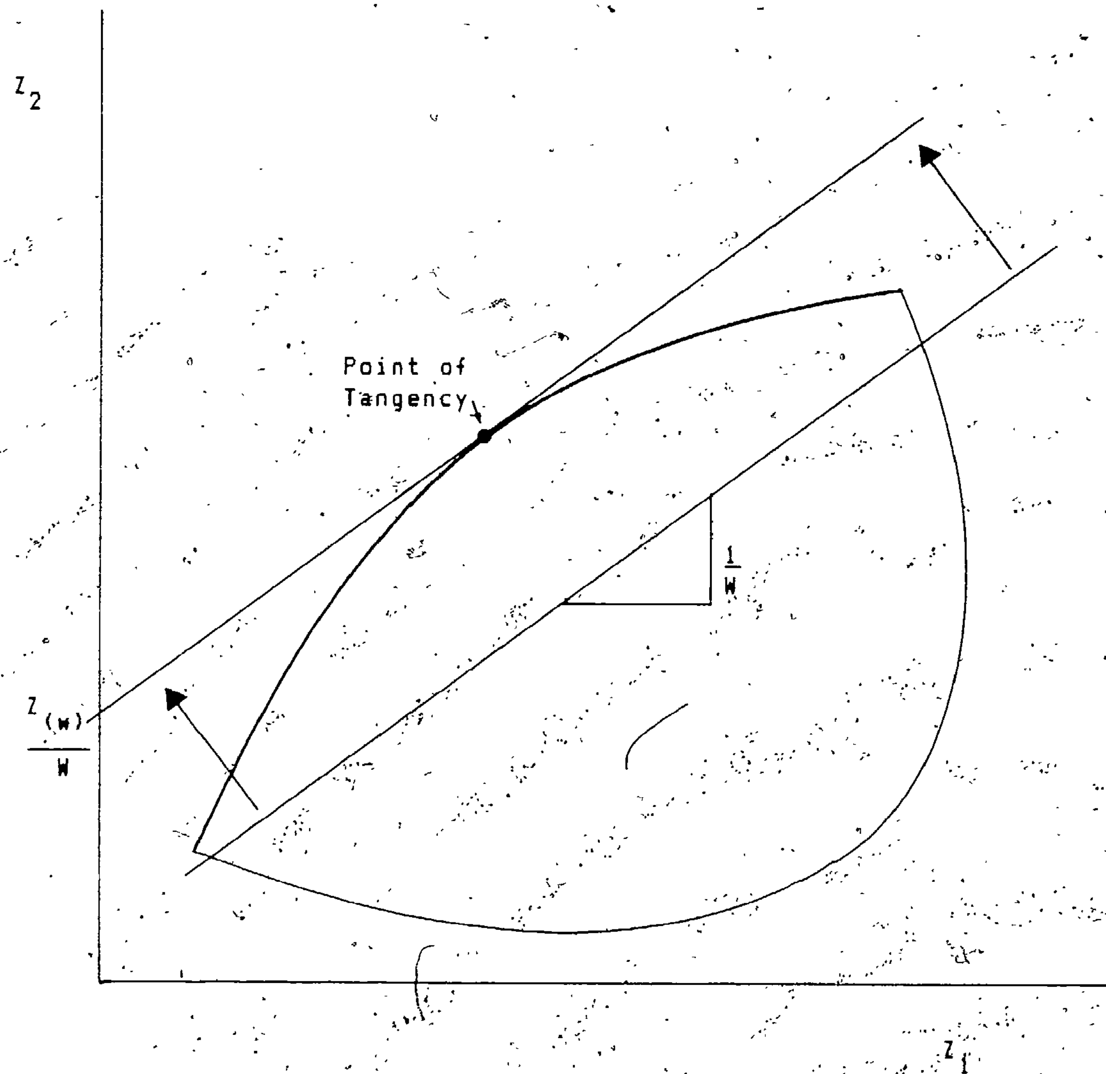
The above example concerns a multiobjective research problem where two objectives are to be maximised. A number of multiobjective research problems exist where one of the objectives is to be maximised while keeping another objective to a minimum, as for example in the MVP and TDVP problems. These types of problems can be shown to be similar to the one discussed previously, except that the indifference lines are not pushed in a northeasterly, but in a northwesterly direction (see Figure 3.3).

3.4.2. The Constraint Method

The constraint method transforms a multiobjective problem into a number of single objective problems by optimising for one objective while constraining all the other objectives to specified values. This method was first suggested by Marglin (1967) and has been applied for example by Cohon and Marks (1975).

Let us return once more to Schilling's fire station location problem. Suppose another constraint was added into the constraint set, one which specified that at least n people had to be covered within a specified distance of the fire station to be located. The additional constraint reduces the feasible region of the objective space (see Figure 3.4). The question then becomes that of how much property value

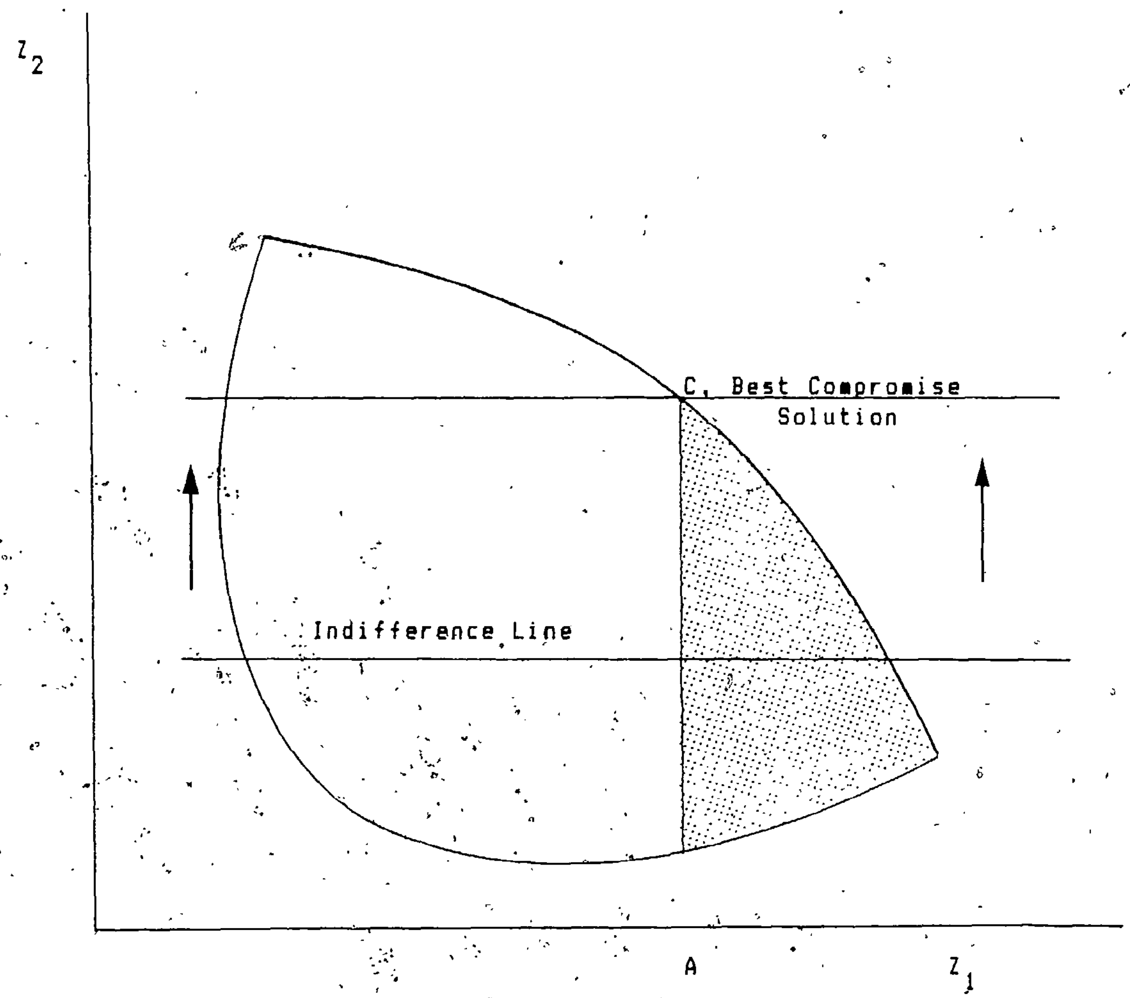
Figure 3.3 The Weighting Method Applied to a Multiobjective Problem where One Objective is to be Maximised while keeping the other to a Minimum



= slope

= the indifference line is pushed <u>Northwest</u> until it is tangent to the noninferior solution set

Figure 3.4 Generating a Noninferior Solution Set Graphically by the Constraint Method



= area that is no longer part of the feasible region of the objective space

A = maximum constrained imposed on Z_1

→ = the indifference line is pushed North until a further gain in Z_2 will imply a loss in Z_1.

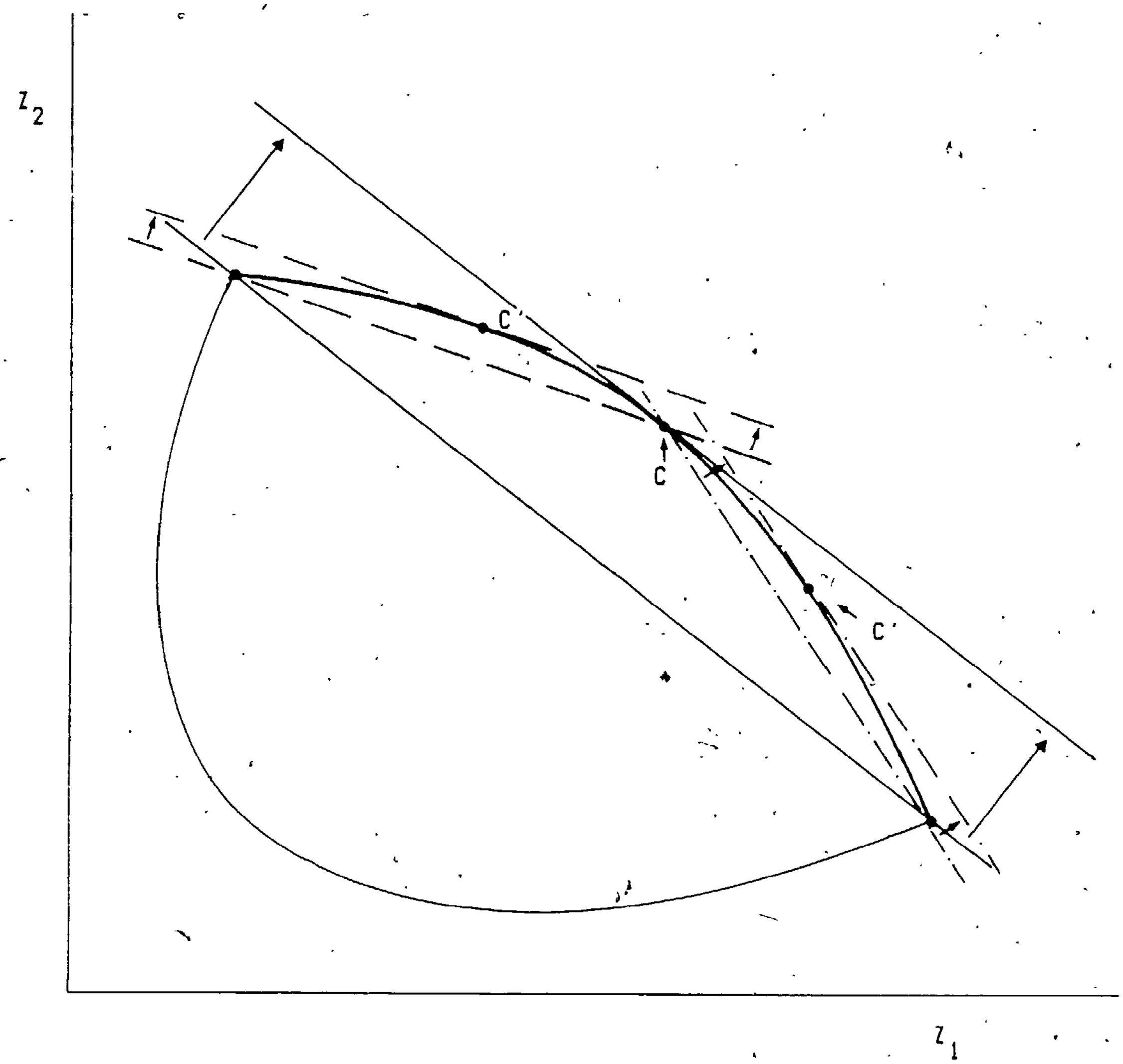
can still be covered without violating the imposed population constraint. Defined this way, the problem involves indifference lines that are parallel to the constrained objective function, and the best compromise solution for the given added constraint (point C in Figure 3.4) can be identified by pushing the indifference line north until a further gain in the unconstrained objective will imply a loss in the constrained objective.

The endpoints of the noninferior solution set are obtained by optimising each of the two objectives individually. The curve between the two points is evaluated by stepwise increases in the constraint imposed on one of the objectives, each time evaluating for the point of maximum utility, or the best compromise solution. The number of points that will approximate the noninferior solution set will depend on the size of the increase of the added constraint on each step.

3.4.3. The Noninferior Set Estimation Method (NISE)

The noninferior set estimation method, hereafter referred to as the NISE method, is one developed by Cohon (1978). The idea underlying this method is to allow for a quick convergence onto a good approximation of the noninferior solution set. The method operates by first identifying the endpoints of the solution set as noted previously. Given a convex noninferior solution curve, the line segment between the two extreme points must comprise a set of the feasible region (see Figure 3.5). This line segment can be treated as an indifference line. The northeast rule can now once again be applied to this indifference line, forcing the line northeast until it becomes tangent to the noninferior solution

Figure 3.5 Generating a Noninferior Solution Set Graphically by the NISE Method



——— = indifference line connecting the extreme points

C = first point of tangency

— — = indifference lines connecting the extreme points to the first point of tangency

C' = subsequent points of tangency

——→ = the indifference line is pushed <u>Northeast</u> until it is tangent to the noninferior solution set curve, thereby identifying one of the points on the noninferior solution set.

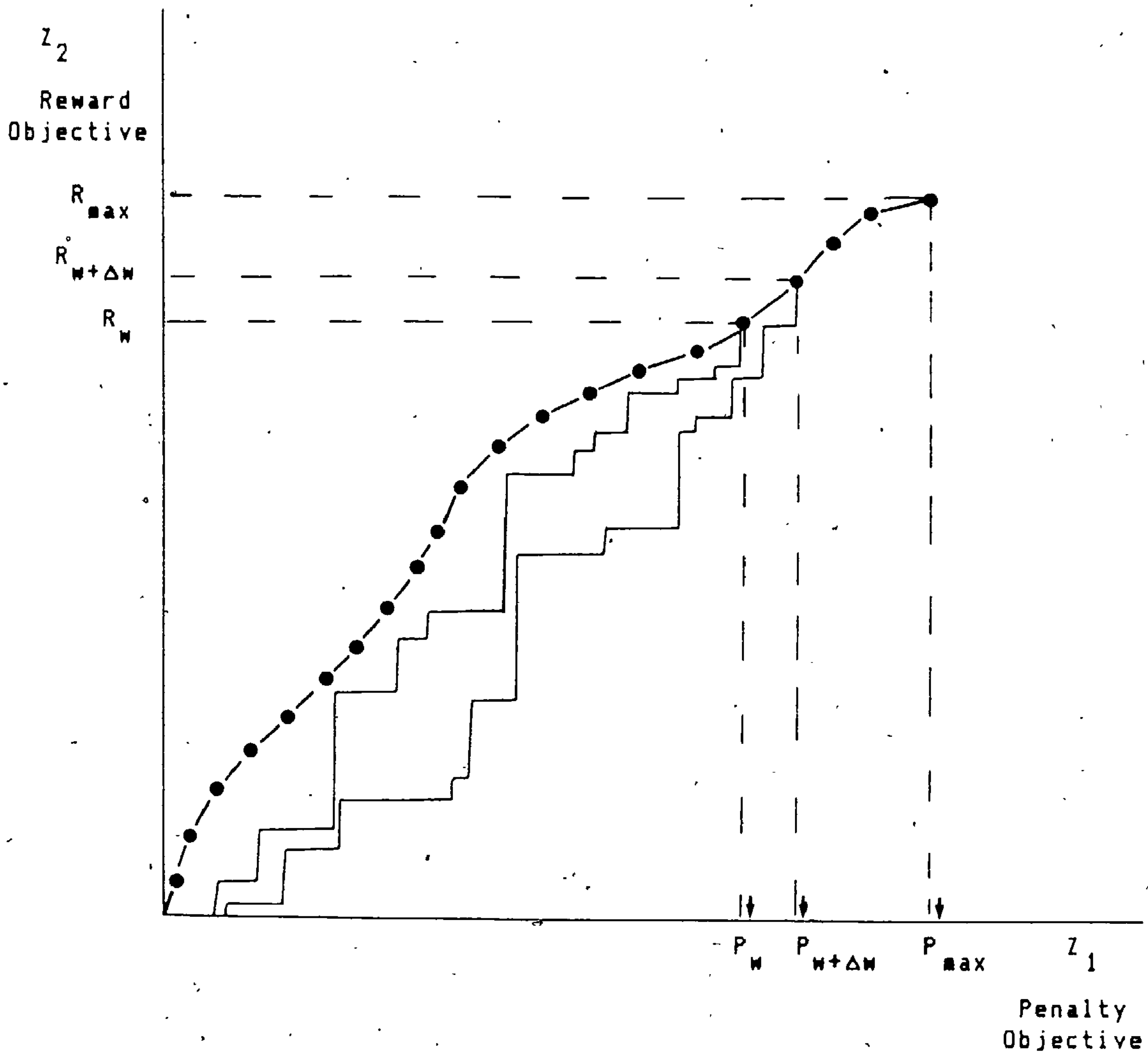
Subsequently, the slopes for two new line segments can be identified, those of the two line segments that connect the extreme points to the last discovered point of tangency. Treating these two lines as two new indifference lines, and again applying the northeast rule, two further points of the noninferior solution set can be identified. The method can be repeated again and again for each new set of possible line segments, resulting in an increasingly accurate approximation.

3.5. Choosing an Estimation Technique

Cohon notes that the choice of technique to use when evaluating a multiobjective optimisation problem depends on the analyst's preference for mathematical procedures, the nature of the noninferior solution set and the type of results expected from the analysis (Cohon, 1978:156). Selecting a technique to solve MVP or TDVP type problems, therefore, first requires a closer inspection of the nature of their underlying noninferior solution sets.

These noninferior solution sets will be demonstrated later on to consist of a set of discrete points. Each discrete point is identified by the total reward and the total penalty associated with a noninferior route sequence. A hypothetical example of some noninferior route sequences and their associated discrete noninferior solution points are shown in Figure 3.6. A hypothetical trade off curve or noninferior solution curve derived when connecting the set of all noninferior discrete points is shown. It will be demonstrated in Chapter Six that this trade off curve is not convex. A convex set has the following

Figure 3.6 Example of an MVP or TDVP Noninferior Solution Set



● = some noninferior solutions identified by discrete points

●—●—●—● = approximation of the noninferior solution curve

(P_{max}, R_{max}) = discrete solution point for the most optimal route when connecting all possible nodes at least penalty

(P_w, R_w) = discrete solution point for the most optimal route when total penalty $\leq P_w$

$(P_{w+\Delta w}, R_{w+\Delta w})$ = discrete solution point for the most optimal route when total penalty $\leq P_{w+\Delta w}$

= the actual path of a route; links are travelled when the path is horizontal, nodes are serviced when vertical property: Points on a line connecting any two points in that set are also members of that set. A closer inspection of Figures 6.2 and 6.3 in Chapter Six shows that this property is violated since the noninferior solution sets underlying MVP and TDVP type problems may consist of stepwise increases.

Violation of the convex property rules out the possibility of utilising the NISE method as a possible solution technique since, as noted previous, a convex noninferior solution set is a necessary assumption for this method.

Two different types of output may be expected from an MVP or TDVP analysis. First, the analyst may wish to evaluate for the solution that maximises reward given a maximum acceptable penalty threshold, or the solution that minimises penalty given a minimum expected reward threshold. This type of analysis will result in the identification of one member of the noninferior solution set. Second, the analyst may wish to evaluate for a subset of, or the entire noninferior solution set.

The question that remains is which of the two noninferior solution set generating techniques, the weighting method or the constraint method, is the one most appropriate for the two types of analysis outlined above.

The strength of the weighting method lies in its applicability to problems where the weights themselves are of some importance in the interpretation of the results. This is not necessarily the case here. The weighting method also suffers from the drawback that it can give poor coverage of the noninferior solution set by getting stuck in the vicinity of one of the endpoints, or in a small range somewhere within the noninferior solution set. This potential skipping over large portions of the noninferior solution set tends to be beyond the analyst's control, but could be overcome by choosing different weights with different specified uniform increments on the basis of trial and error. This is however arkward and time consuming. Given that the weighting method does not necessarily guarantee a good coverage of all the noninferior solution set, and given that the method is not very appropriate for the identification of merely one specific point on the solution set, it is rejected as a solution technique.

The constraint method generally results in a reasonable coverage of the noninferior solution set since it operates by constraining one of the two objectives at set intervals. It is therefore the most appropriate approach if a good overall approximation of the noninferior solution set is requested. Also, the constraint method is ideal for the identification of a single specific point on the curve since this identification in itself depends on constraining one of the two objectives. For these reasons, the constraint method is thought to be the most appropriate approach to solving MVP and TDVP problems. It is therefore the technique utilised in this study.

CHAPTER FOUR

4.0 Mathematical Definitions of the MVP and TDVP Problems and Possible Solution Approaches

4.1 Introduction

The objectives of this chapter are to define the MVP and TDVP problems mathematically, and to discuss possible solution approaches. The chapter sets out by defining the MVP problem in two ways, as a zero-one integer programming problem and as a route sequencing problem. Subsequently, the TDVP problem is defined as a route sequencing problem.

The discussion of possible solution approaches commences by assessing the applicability of exact solution procedures. Subsequent discussion concerns the possible applicability of a heuristic.

4.2 Mathematical Definitions of the MVP Problem

Briefly recapitulating, the MVP problem in its most general form concerns a set of N nodes or demand points, each associated with a known potential reward if visited. The demand nodes are connected by links, and a penalty of known magnitude is associated with travelling these links. The overall objective is to travel to a number of demand points in an effort to maximise reward, while at the same time keeping the penalty accrued for travelling the links connecting the nodes to a minimum.

The problem is multiobjective in nature for two reasons. First, no relationship is defined between reward and penalty, and the two objectives are therefore treated as non-commensurable. Second, it is not assumed that all of the N nodes are to be visited. On the contrary, it is assumed that inspection of the trade off curve generated between visiting more nodes and accepting more penalty will allow the identification of a point of optimum return beyond which the marginal increase in reward is outweighed by the marginal increase in penalty.

4.2.1 The Zero-One Integer Programming Definition

The MVP problem, in its most general form, can be expressed as a multiobjective zero-one integer programming problem as outlined below:

$$\text{Maximise} \qquad Z = (Z_1, Z_2) \qquad 4.1$$

$$\text{where} \qquad Z_1 = \sum_{i \in N} R_i \sum_{j \in N} x_{ij} \qquad (\text{ Total Reward }) \qquad 4.2$$

$$\text{and} \qquad Z_2 = \sum_{i \in N} \sum_{j \in N} x_{ij} P_{ij} \qquad (\text{ Total Penalty }) \qquad 4.3$$

subject to:

$$\sum_{i \in N} x_{ij} = 0 \text{ or } 1 \qquad j \in N \qquad 4.4$$

$$x_{ij} = (0,1) \qquad j \in N \qquad 4.5$$

$$\sum_{i \in N} x_{ij} - \sum_{k \in N} x_{jk} = 0 \qquad j \in N \qquad 4.6$$

$$x_{1j} = 1 \qquad 1,j \in N \qquad 4.7$$

$$x_{i1} = 1 \qquad i,1 \in N \qquad 4.8$$

where:

R_i = the reward potential of node i

x_{ij} = 1 if the route involves travel from node i to node j, otherwise 0

P_{ij} = the penalty associated with travelling the link from i to j

1 = the source node and terminal node

N = the set of potential demand nodes

Constraint 4.4 ensures that each node is visited once and once only or is not visited at all. Constraint 4.5 gives the problem the zero-one integer characteristic, ensuring that the link connecting node i with node j is either fully connected or not included in the path. In other words, it ensures that no partial linkages are formed. Constraint 4.6 ensures that every node that is entered is also exited. Constraints 4.7 and 4.8 ensure that node one is both the starting node and the terminal node, turning the path into a closed circuit. The fact that the path resembles a closed circuit is an important consideration when discussing possible solution approaches, a topic elaborated upon in section 4.3 of this chapter. If constraint 4.8 was changed to read

$$x_{id} = 1 \qquad i,d \in N \tag{4.9}$$

where d now stands for the terminal node and $d \neq 1$, then the above problem would turn into a multiobjective shortest path problem, the MPSP problem researched by Current (1981).

The above zero-one integer programming definition permits the formation of subtours. To avoid subtours a large set of additional constraints must be added to the problem formulation, an issue adressed in section 4.4.1 of this chapter.

If, as discussed in the previous chapter, the constraint method is the most preferred solution technique for solving the above problem, then it is necessary to add a further constraint. An example of such a constraint, one that operates by imposing a threshold on the penalty objective, is shown below:

$$\sum_{i\in N}\sum_{j\in N} x_{ij} P_{ij} \leq PMAX \tag{4.10}$$

where PMAX stands for the maximum possible amount of penalty acceptable.

The above definition assumes that the collection of reward involves no penalty in itself. If penalty is also involved in collecting reward at the different nodes, and if that penalty is proportional to the amount of reward collected, then equation 4.3 and constraint 4.10 need to be adjusted to equation 4.11 and constraint 4.12 respectively.

$$Z_2 = \sum_{i\in N}\sum_{j\in N} x_{ij} P_{ij} + \sum_{i\in N} Q R_i \quad \text{(Total Penalty)} \tag{4.11}$$

and

$$\sum_{i\in N}\sum_{j\in N} x_{ij} P_{ij} + \sum_{i\in N} Q R_i \leq PMAX \tag{4.12}$$

Q stands for the amount of penalty which must be accepted in order to collect one unit of reward.

The mathematical model outlined above defines the MVP problem in its most general form as a zero-one integer programming problem. The MVP problem can alternatively be expressed mathematically as a route sequencing problem.

4.2.2 The Route Sequencing Definition

The mathematical definition of the general MVP problem in the form of a route sequencing problem is shown below:

Maximise $$Z = (Z_1, Z_2)$$ 4.13

where $$Z_1 = \sum_{i=1}^{m} R_{S_i} \quad \text{(Total Reward)}$$ 4.14

and $$Z_2 = \sum_{i=1}^{m} P_{S_i S_{i+1}} \quad \text{(Total Penalty)}$$ 4.15

subject to:

$$S_i \in \{1,\ldots,N\} \quad \text{for all } i$$ 4.16

$$S_i \neq S_j \quad \text{for all } i,j \; ; \; i \neq j$$ 4.17

$$S_1 = 1$$ 4.18

$$S_{m+1} = 1$$ 4.19

Here, m represents the total number of nodes that are visited in the tour specified. S_i, the ith member of the route sequence vector $\underline{S}$, stands for the node visited on the ith step of the tour. R_{S_i} stands for the reward to be collected at the node identified by S_i on the ith step of the tour, and $P_{S_i S_{i+1}}$ stands for the penalty that has to be accepted for travelling the ith link of the tour. Condition 4.18 ensures that S_1, the first node on the tour is the actual starting point, the depot. Constraint 4.19 ensures that, after the last node on the route has been visited, the route connects back to the starting node, enforcing the condition of a closed circuit. Condition 4.17 ensures that no node is visited more than once and condition 4.16 ensures that all the nodes visited on the route are members of the set of all nodes, N.

The imposition of a threshold constraint on the penalty objective for this problem definition would be the following:

$$\sum_{i=1}^{m} P_{S_i S_{i+1}} \leq PMAX \qquad 4.20$$

Finally, with a penalty component for the actual collection of reward, objective 4.15 and constraint 4.20 need to be rewritten as follows respectively:

$$Z_2 = \sum_{i=1}^{m} P_{S_i S_{i+1}} + \sum_{i=1}^{m} Q\, R_{S_i} \quad \text{(Total Penalty)} \qquad 4.21$$

and

$$\sum_{i=1}^{m} P_{S_i S_{i+1}} + \sum_{i=1}^{m} Q\, R_{S_i} \leq PMAX \qquad 4.22$$

4.3 Mathematical Definition of the TDVP Problem

As already noted in the introductory chapter, the TDVP problem is a more complex version of the MVP problem. Demand potential at the different nodes is no longer assumed to be independent of the time of arrival, but to depend on some time specific demand function.

The actual time of arrival at the different nodes to be visited therefore becomes an important consideration in the problem evaluation. The time of arrival at a node i, A_{S_i}, can be expressed as follows:

$$A_{S_i} = A_{S_{i-1}} + t_{S_{i-1}S_i} + F_{S_{i-1}} \quad \text{for all } i,\ i=2,\ldots,m \qquad 4.23$$

where:

A_{S_1} is the time of first arrival at the depot

S_i, for all i, i=1,...,m is the identity of the ith node visited on the ith step of the tour where m is the number of nodes visited on this particular tour.

$t_{S_{i-1}S_i}$ stands for the time required to travel from S_{i-1} to S_i.

$F_{S_{i-1}}$ is the time spent collecting the reward at node S_{i-1}.

Given this definition of arrival time, the TDVP problem can now be defined, as a route sequence problem. To simplify the problem formulation, a number of assumptions are made. The first is that the same time dependent demand function applies to all nodes. A second assumption is that demand potential depends only on the time of arrival, and does, therefore, not fluctuate throughout the vendor's stay. These two assumptions are discussed in more detail in Chapters Seven and Eight. A route sequence definition of the TDVP problem is given below.

Maximise $Z = (Z_1, Z_2)$ 4.24

where $Z_1 = \sum_{i=1}^{m} REX_{S_i A_{S_i}}$ (Total Reward) 4.25

and $Z_2 = \sum_{i=1}^{m} P_{S_i S_{i+1}}$ (Total Penalty) 4.26

subject to:

$S_i \in \{1,\ldots,N\}$ for all i 4.27

$S_i \neq S_j$ for all i,j 4.28

$S_1 = 1$ 4.29

$S_{m+1} = 1$ 4.30

Here, $REX_{S_i A_{S_i}}$ stands for the reward that can be expected and collected at node S_i on the ith step of the tour following arrival at time A_{S_i}. Constraints 4.27 to 4.30 are the same as constraints 4.16 to 4.19 when defining the MVP problem as a route sequencing problem. The imposition of a threshold constraint on the penalty objective for this problem definition includes the incorporation of the same constraint as applied to the MVP problem, constraint 4.20.

With a penalty component for the actual collection of reward, constraint 4.22 is again substituted for constraint 4.20. The penalty objective 4.26 will need to be redefined as:

$$Z_2 = \sum_{i=1}^{m} P_{S_i S_{i+1}} + \sum_{i=1}^{m} Q\, REX_{S_i A_{S_i}} \quad \text{(Total Penalty)} \qquad 4.31$$

4.4 Solution Approaches

It was shown in Chapter Three that the multiobjective MVP and TDVP problems can be solved as k single objective problems, where k represents the number of points calculated to approximate the noninferior solution set. The following discussion will focus on how to solve for each of the single objective problems.

It was noted in the introduction that the MVP and TDVP problems concern routes that exhibit the characteristic of a closed circuit, the condition that defines the travelling salesman problem. The MVP and TDVP problems were thus argued to represent special cases of the general travelling salesman problem. The following discussion of possible solution approaches will therefore commence by inspecting solution techniques designed to solve the general travelling salesman problem.

The travelling salesman problem is a member of a general class of routing problems that can be defined as zero-one integer programming problems. Such a mathematical definition is given by Hartley (1976: 436) or Killen (1983: 218).

One of the important characteristics of the general travelling salesman problem is that, for n nodes, there exist (n-1)!/2 potential solutions. This implies that the number of possible solutions increases geometrically with an increase in problem size. An exhaustive search of all possible solutions for a five node problem would, for example, require the evaluation of 12 scenarios. For a 10 node problem this increases to 181,000 possible solutions, and for a 15 node problem the number has risen to 43 billion.

The travelling salesman problem is in actual fact a prototypical example of a special class of difficult combinatorial problem termed NP-hard or NP-complete (Papadimitriou, 1977). These are problems which are so difficult to solve intrinsically that no algorithm has been found to solve them which is efficient in the sense of being a polynomial time algorithm (Papadimitriou, 1977; Garey and Johnson, 1979). A number of less efficient algorithms have however been devised to solve these types of problems, including the travelling salesman problem (see for example Bellmore and Nemhauser, 1968; and Bodin et al., 1983). The following section will discuss some of these algorithms which are potentially applicable to solving the MVP or TDVP problems.

4.4.1 Exact Solution Techniques

The possible mathematical definition of the travelling salesman problem, as well as the MVP problem, as a zero-one integer programming problem has been demonstrated. One of the most commonly employed techniques for solving zero-one integer programming problems is by a special version of general linear programming. Techniques for solving general linear programming problems, such as by the simplex method, are discussed in detail by Wagner (1969) and Killen (1979, 1983).

General techniques for solving linear programming problems may yield real value solutions. The solutions to the problems of interest here are however confined to integers, and thus only a finite set of discrete points are feasible answers. Any general technique for solving a linear programming problem must therefore be supplemented by some search method that is capable of identifying the set of discrete points within the solution space, and evaluating amongst these for the one that is optimal. One such procedure is the method of cutting planes developed by Gomory (1958). Another is Dakin's branch and bound algorithm (Salkin, 1975).

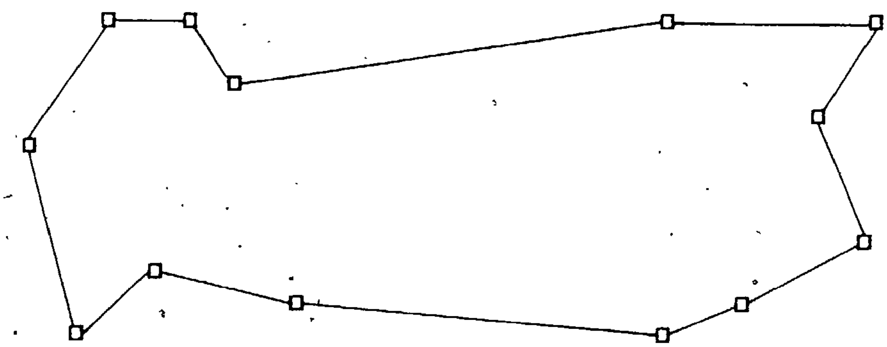
A major drawback when utilising such an adapted technique of linear programming is the possible formation of subtours in the solution resulting in disjoint circuits (see Figure 4.1). Any solution containing subtours is of course not an example of one closed circuit. Additional constraints must therefore be introduced into the problem formulation to ensure this single closed-circuit characteristic (Dantzig <u>et al.</u>, 1954; 1959; Bellmore and Malone, 1971 and Scott, 1971). Even for relatively small problems, say concerning five nodes, there exists a very large number of possible subtours, and the number of additional constraints which must be introduced into the problem formulation is very great.

Given the complexities that have to be added to a general linear programming technique when attempting to solve a travelling salesman problem, the capacities of most contemporary computers are exceeded when attempting to solve all but the smallest of problems. This type of a solution approach is therefore confined to the smallest of travelling salesman problems, say less than ten nodes, and is therefore not of much use when attempting to solve the more complex MVP and TDVP problems. This is especially true since MVP or TDVP problems may concern a large set of possible demand points.

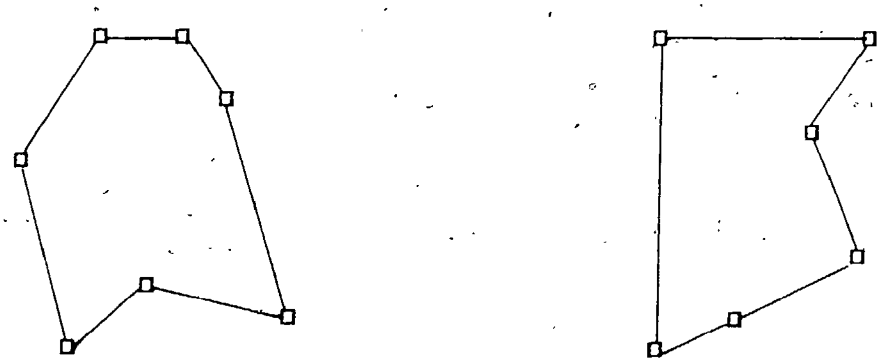
Figure 4.1 Examples of a Closed Circuit and Disjoint Circuits

□ = Nodes

Closed Circuit:



Disjoint Circuits:



A somewhat more efficient and practical approach to solving MVP or TDVP problems via an exact method is by some tree searching procedure (Kruskal, 1956). Wagner (1969), discussing combinatorial programming, demonstrates the method of excluded subtours and the method of partial subtours. The method of excluded subtours commences by defining a possible solution to a problem by inspection. The obtained solution is regarded as an upper bound to the problem. This problem is now solved as an assignment problem, and the result is tested for subtours. If no subtours result, then the solution arrived at by inspection is the optimal solution. If subtours emerge, branching is initiated on the smallest subtour. Of the resulting solutions, the one which yields the lowest value below the previously set upper bound is substituted as the new upper bound. The procedure is repeated until no solution comprising subtours for which the results are lower than the last defined upper bound (Killen, 1983; 220). The solution thus derived is known to be the optimal solution (Killen, 1983: 220). This method involves an assignment linear programming problem at each node of the combinatorial tree (Killen, 1983: 220), a situation avoided in the method of partial tours.

The method of partial tours again involves the determination of an initial solution by inspection, and the declaration of the associated solution as the upper bound. Selecting one specific node, such as the starting node, the characteristics underlying a closed circuit imply that all but one out of all possible links to the n-1 remaining nodes must equal zero, the remaining node taking on the value of one. Determining sub-solution values for all n-1 possible branches from the starting node by a technique similar to that of the Vogel Approximation Method (Killen, 1983: 58-60), the link to the lowest sub-solution value is assigned unity, that is, it is connected to the starting node. Subsequently the newly connected node is branched to all remaining n-2 nodes and again, the node with the lowest possible sub-solution value is connected. The process is repeated for n-2 branchings, at which point a complete route with an associated solution value can be identified. This solution value is set as the new upper bound and the entire process is repeated until no new solution is found that yields a value lower than the previous upper bound.

Again, it is clear that for problems which involve a very large number of nodes, these solution methods become unmanageably large, since it is necessary either to utilise linear programming, incorporating a very large constraint set, or since the number of solutions to be retained in the computer memory at any one time become so large that they will exceed the capacity of the computer. Examples of exact algorithms of this type are found in Held and Karp (1971), Knuth (1976), Karp (1977) and Crowder and Padberg (1980).

The above examples, as well as a more lengthy discussion of the topic by Bodin <u>et al.</u> (1983) suggest that, given existing computational capacities, exact solution techniques to solving the general travelling salesman problem utilising linear programming or tree search procedures are computationally too burdensome to solve all but the smallest of problems. Since research undertaken in mathematics and operations research has not yet found other efficient exact solution methods, it is unlikely that this solution method, given existing constraints of computer capacities, will offer a feasible approach to solving the more complex MVP and TDVP type problems. A compromise will therefore have to be reached between the quality of the optimal solution to be obtained and the total expenditure of computational effort needed to obtain that solution. Such a compromise can be reached by turning to heuristic solution approaches.

4.4.2 Heuristic Solution Approaches

Scott (1971: 55) notes that heuristic programming is a technique which dispenses with rigour and exactness, and in exchange is characterised by flexibility and practicality. Heuristic programming, more of a general problem solving philosophy than a rigidly defined mathematical procedure, basically involves deriving some solution to a given problem by means of a set of rules (Scott, 1971: 50). These sets of rules range from a process of trial and error to an elaborate computer search where repeated locally optimised iterations will result in convergence of a solution in the direction of the overall optimal solution. Heuristic algorithms operating on the principle of iterative local optimisation are generally termed steepest ascent n point move algorithms (where n = 1,2,3,... and defines the stepsize of any move at any one iteration). Given sufficient iterations, the convergence of a solution in the direction of global optimality may lead to a final solution which is close to or at the optimal solution. The margin of potential departure from absolute optimality is however not known. There also exist types of problems where the solution space includes local optima. In this case the converging solution may get locally trapped.

Utilising a heuristic algorithm to solve the general travelling salesman problem appears to be a widely accepted approach, and a great variety of such heuristics have been developed. For an in-depth discussion of these heuristics the user is referred to Bodin et al. (1983).

Summarised briefly, these heuristic programs tend to involve one, or a combination of two processes. One process involves tracing out a continuous path through a diminishing or growing sequence of solutions. Beginning with some arbitrary and usually feasible solution, the algorithm works towards optimality by locally optimising expansions or contractions of that solution set. An example here is the Karg and Thompson algorithm (Karg and Thompson, 1964), a steepest ascent one point move algorithm. Selecting and connecting a pair of starting points, a third point is searched for which is the closest to either of one of the two end points. This point is now connected. By successive iterations all points are thus connected to the respective nearest end point of the incomplete circuit until, ultimately, only two endpoints remain to be closed.

A second algorithm, devised by Cooper (1968), also an example of a steepest ascent algorithm, involves a shuffling process where some arbitrary and feasible solution is successively restructured by changing the relative positioning of the elements until no further improvements to the solution can be found.

A general distinction in heuristic algorithms is that between a forward and a backward approach. Usually applied to optimal network problems, the forward method starts with some infeasible solution, such as the null solution, and works forward by stepwise arc addition and a shuffling process. The backward method starts with some feasible solution, usually the complement to the null solution and works backwards by stepwise deletions and a shuffling process. The shuffling process, for example the subroutine Netran (Scott, 1971: 50), is an iterative process of arc deletion and reassignment, or link restructuring, so as to gain maximum improvement in the value of the objective function. Examples of research attempting to solve the travelling salesman problem via a heuristic, utilising procedures such as those outlined above, are Bellmore and Nemhauser (1968), Lin and Kernigham (1973), Gupta (1978) and Golden et al. (1980).

A somewhat different heuristic approach for solving very large travelling salesman problems is that by Litke (1984). Attempting to solve the manufacturing engineering problem of sequencing the path of a drill on a circuit board, Litke devised what is basically a 'divide and conquer' approach. This approach involves gathering points that are close together into clusters, where each cluster must contain less than a specified number of points. Subsequently, an exhaustive search is applied to identify the optimal path between and within the different clusters. This heuristic approach certainly appears to be an improvement on the more traditional 'band method' used to solve this type of problem. The method does however rely on a visual inspection of the data set and a visual clustering of the points. It can therefore only be as good as the operator's capacity to "see where the points are" (Litke, 1984: 1229).

Several other research approaches have used non-exact methods for solving large travelling salesman problems. One such approach adopts a probabilistic viewpoint, as for example in Beardwood et al. (1959), Eilon et al. (1971), Karp (1977) and Stein (1978). Given that nodes are reasonably evenly distributed over a region, the probabilistic approach will often provide a good approximation of the overall optimal length of the tour, but will not identify the tour itself. This solution approach is therefore of little interest beyond such an approximation, and is not really applicable to the types of problems which are of interest in this study.

4.5 Conclusion

This chapter set out by showing that MVP and TDVP type problems can be defined mathematically as both multiobjective zero-one integer programming problems and as multiobjective route sequencing problems. Chapter Three demonstrated that these multiobjective problems can be solved as k single objective problems. Each of these single objective problems represents a more complex version of the general travelling salesman problem. It has been demonstrated that it is presently computationally infeasible to solve large general travelling salesman problems by an exact solution approach. This was therefore rejected for the more complex MVP or TDVP problems. Successful applications of heuristic programming to the general travelling salesman problem were discussed, and different heuristic strategies were outlined. The following chapter will concern itself with the design of a heuristic that is capable of solving MVP problems.

CHAPTER FIVE

5.0 The MVP Program Design

5.1 Introduction

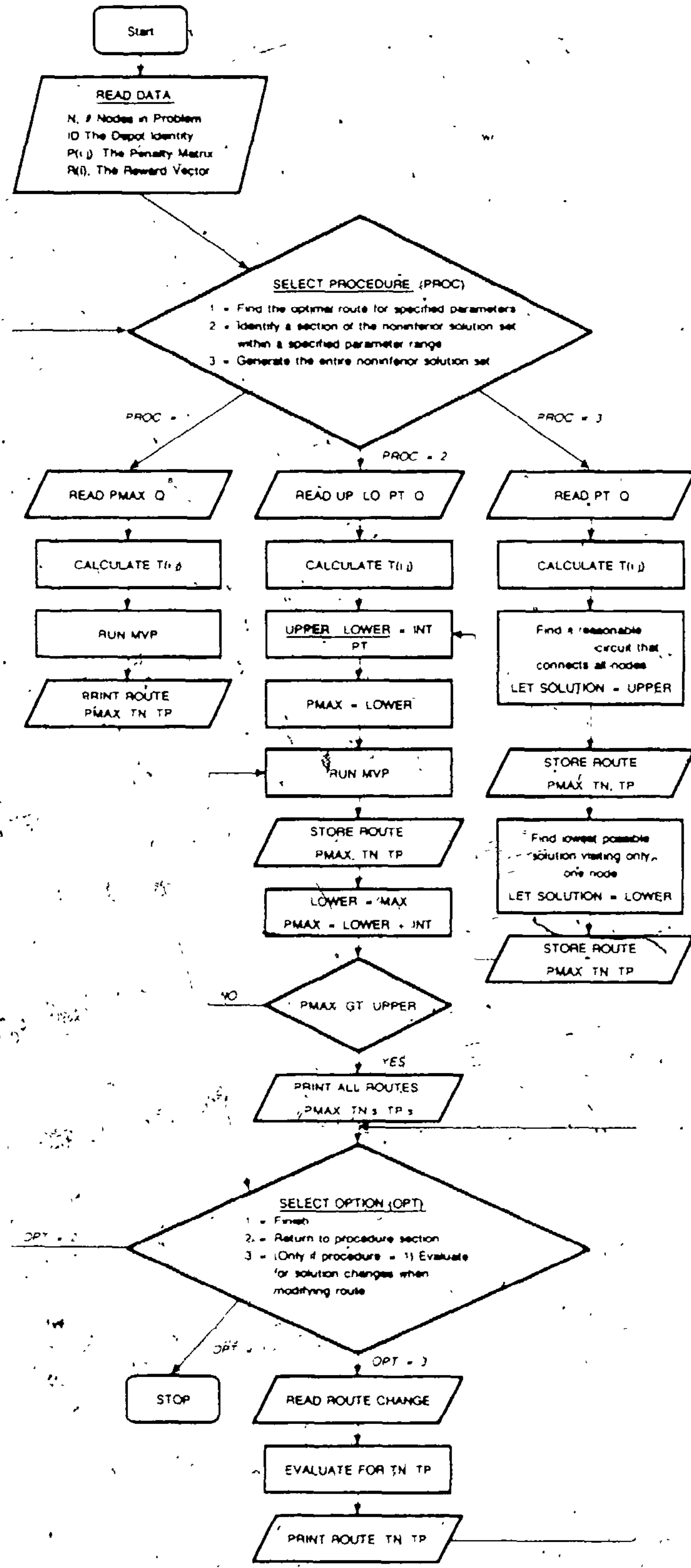
Chapter Three showed how a noninferior solution set could be approximated by evaluating k members of that set. The constraint method was identified as the one most suitable for solving MVP and TDVP type problems. It was noted that the evaluation of each of the k members involves solving a single objective problem. Chapter Four assessed the feasibility of a number of different solution approaches to solving each of these single objective problems. It was concluded that a heuristic poses the most feasible solution technique. Chapter One outlined the advantages to interactive programming. An interactive program to solve MVP type problems, based on the constraint method, has been designed and is described in some detail in this chapter. The objective to be constrained is the link penalty. The program is described in two parts. Part one outlines the overall design, while the second discusses the logic and procedures of the actual heuristic.

5.2 The MVP Program

The design of the MVP program is summarised in the flowchart shown in Figure 5.1. The different steps are described below.

The program commences by reading the input data from an input file, including the number of nodes, N, considered in the problem, the identity of the depot, ID, and the penalty matrix for travelling between the nodes, [P], of dimension, N*N. This is followed by the names and reward potentials of the different nodes, vectors R and NAME of length

Figure 5.1. The MVP Program Design



N. Having read the input data, the program will prompt the user to select one of three possible choices for proceeding with the analysis. The choices concern the type of analysis desired.

Selecting Procedure One will result in the identification of one optimal solution for a set of specified parameters. The program will prompt for a threshold value for the penalty objective which is not to be exceeded, PMAX, and for, Q, the penalty that must be accepted to collect one unit of reward. If the reward can be collected instantaneously, then this parameter is set to 0. Given a value for Q, the program will calculate a matrix (t) of dimension N*N. This matrix represents the sum of the penalty that must be accepted to travel from node i to node j and the penalty for collecting the reward at node j.

Thus: $$t_{ij} = P_{ij} + Q(R_j) \quad \text{for all } i,j; \quad i,j=1,\ldots,N \qquad 5.1$$

The program subsequently calls the MVP heuristic, which will search for that subset of nodes, and the associated route sequence, that will yield maximum reward while remaining within the penalty threshold imposed by PMAX. The output will show the optimal identified route sequence, total reward collected, and total penalty accrued.

Selecting Procedure Two allows the user to evaluate for a window of the noninferior solution set. The program will again prompt for a specification of Q. It will further prompt for an upper and a lower threshold for the penalty objective, and for the number of points to be calculated to approximate the window of the noninferior solution set. The program will again calculate the matrix (t). It subsequently searches for the optimal solution when constraining total penalty to not exceed the lower defined threshold. Once the optimal solution has been identified, it is stored in memory, and the penalty constraint is increased by a calculated increment, derived by dividing the difference between the lower- and the upper-defined penalty threshold by the number of points specified to approximate the window. The program will continue to increment the penalty threshold, and to search for the associated optimal solution until PMAX exceeds the upper-defined threshold. The output for Procedure Two will show the route sequences, the associated total rewards, and the associated total penalties of all the solutions found.

Procedure Three allows the user to derive an approximation of the entire noninferior solution set for a specified value of Q. The user will be prompted to specify how many points are to approximate the noninferior solution set. Having calculated {t}, the program will identify the intervals on the penalty objective that are to be evaluated. Commencing with PMAX = 0, the program will repeatedly call the MVP heuristic to search for the most optimal solution, stepwise increasing the maximum penalty constraint.

Whatever procedure has been selected, the program ultimately prompts the user on how to proceed further. Three options are presented. Option One is to finish and exit the program. Option Two is to return to the procedure selection stage, and to run a subsequent analysis, say for a different value of Q. Option Three is only applicable if Procedure One was called previously. This option allows the user to modify an optimal route found by Procedure One to incorporate personal preferences or considerations not evaluated by the heuristic. For any specified route modification, the program will show the adjusted route sequence and will calculate the new values of maximum reward and total penalty accrued.

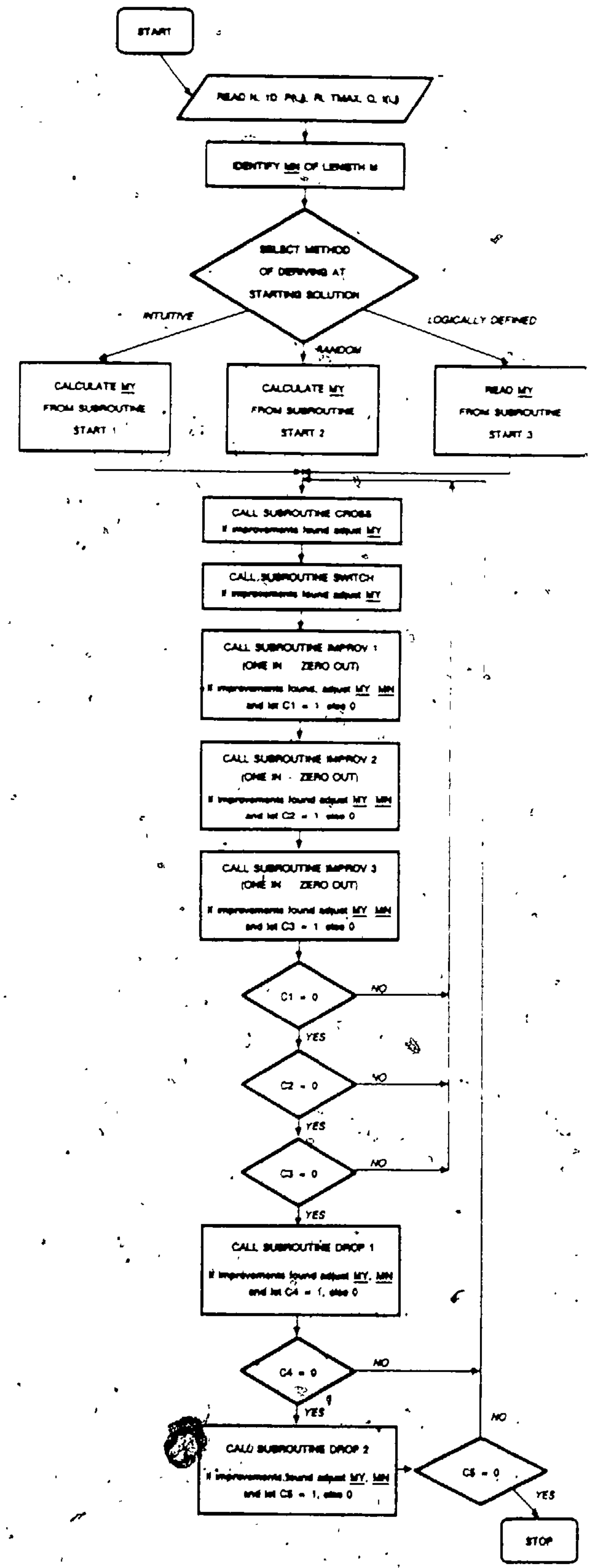
5.3 The MVP Heuristic

The exact procedure underlying the MVP run, the heuristic that attempts to identify the optimal route for a given set of parameters, is described below. The design of the heuristic is the result of experimentation. A number of strategies for detecting possible improvements over a given route sequence were designed and tested. Different orders of implementing the improvement searches were compared. The heuristic described below represents the most satisfactory combination found. The sequence of logical steps or operations underlying the heuristic is presented in the form of a flow chart in Figure 5.2.

For every individual run of the heuristic, the following input must have been specified: Parameters N, ID and PMAX, the matrices (P) and (t) and the vector $\underline{R}$. The first step of the heuristic is to try and reduce the complexity of the overall problem by attempting to reduce the number of nodes to be considered, and to store all the nodes that can be visited in a vector. The strategy is to identify any node that cannot be visited and serviced directly from the depot without violating the maximum penalty threshold, PMAX. This is achieved by testing each node for the following condition:

$$\text{Test if} \quad t_{di} + t_{id} \leq \text{PMAX} \qquad \text{for all } i,\ i=1,\ldots,N \quad d=\text{depot} \tag{5.2}$$

Figure 5.2. The MVP Heuristic Procedure



Every node that satisfies this condition is identified sequentially in vector MN; all other nodes are rejected. If n nodes satisfy the condition, then the vector MN will finally be of length n. This step is especially useful for very small values of RMAX since it allows only those nodes within close proximity to the depot to be considered as feasible nodes.

The heuristic next prompts the user to select one of three possible techniques to generate a starting solution, START1, START2 and START3. The three methods are an intuitive, a random and a logically defined starting approach respectively. Each of these approaches is described in some detail in section 5.4. Whichever approach is used will result in the identification of a starting solution identified by the route sequence vector MY of length m. The origin and destination, O_i and D_i at any one stage i within the route can thus be identified from MY. For example MY_1 identifies the depot, ID. MY_2 represents D_1 and O_2, the destination of link one and the origin of link two respectively. MY_3 represents the destination of the second link, D_2, and the origin of the third link, O_3, ... MY_m, the last entry in the route sequence vector, represents the node visited just prior to returning to the depot. MY_{m+1} will therefore equal MY_1, the depot. With the exception of MY_1, all members that have entered the route sequence vector must, by definition, have previously been members of vector MN. As soon as a node is incorporated in the route, it is removed from MN and added as a member of MY. Once an initial starting route has been identified by MY of length m, the length of MN is thus reduced to n-m. Every time a node is added or dropped from the route, it is transferred between the two vectors, reducing or increasing their respective lengths.

Once a starting solution has been identified, the heuristic proceeds by calling two routines, CROSS and SWITCH. Both routines attempt to swap members within the route sequence vector MY in order to reduce the penalty accrued for travelling the links. These two routines do not alter the set of nodes contained in MY, and therefore do not alter maximum reward collected. The objective of routine CROSS is to search for self crossing paths in the tour. If a self crossing path is detected, then the routine swaps members within MY to eliminate the self cross. The objective of routine SWITCH is simply to shuffle members within MY to search for possible reductions in the length of the path. The two routines are described in some detail in section 5.4.

The heuristic next calls three improvement routines, IMPROV1, IMPROV2 and IMPROV3, which search for possible ways of increasing total reward to be collected while remaining within PMAX. The improvement searches are based on the principles of node addition, substitution and subtraction respectively. These routines are also described in detail in section 5.4. If an improvement can be identified by any one of the three routines, then the heuristic returns to implement routines CROSS and SWITCH, and proceeds by once again calling the three improvement routines. If neither of the five routines should identify a possible improvement in straight succession, then the heuristic moves on to try and identify improvements using two further routines, DROP1 and DROP2, which are also explained in some detail in section 5.4. If either one of these two routines should identify an improvement, then the heuristic returns to routines CROSS and SWITCH, and proceeds by repeating the entire search as before. If routines DROP1 and DROP2 can also not detect any further improvements, then the MVP run returns to the overall MVP program, and the optimal route sequence found is stored or printed.

The remainder of this chapter discusses the different routines mentioned above in more detail.

5.4 Description of the Routines

5.4.1 Identification of the Starting Solution

The MVP heuristic allows the user to select amongst three different starting routines to generate an initial solution. These are routines START1, START2 and START3, based on an intuitive, a random and a logically defined approach respectively. The three routines are described in some detail below.

5.4.1.1 START1: The Intuitive Starting Solution

The intuitive starting method operates by commencing from the depot, and successively asking the operator to specify the next node to be visited on the route. After each entry, a test is performed to determine if the route sequence is still within the maximium penalty constraint imposed, based on the following condition:

$$\text{Test if} \quad \sum_{i=1}^{m} t_{O_i D_i} \leq PMAX \quad , \quad D_m = O_1 \qquad 5.3$$

As soon as this condition is violated for the first time, the most recently added node is rejected and the user is informed which nodes can still be added while remaining within PMAX. The user will then continue selecting nodes to be inserted in the route from those that are still feasible within PMAX until no more feasible nodes remain. The starting route has been found.

5.4.1.2 START2: The Random Starting Solution

To obtain a random starting solution, it is first necessary to standardise the rewards of all nodes presently members of MN so that their sum equals unity. This is achieved by the following calculation:

$$SR_i = \frac{R_i^{\beta}}{\sum_{i=1}^{n} R_i^{\beta}} \qquad \text{For all } i \in \underline{MN} \tag{5.4}$$

SR_i stands for the standardised value of R_i. The constant β allows the probability of selecting nodes with larger or smaller rewards to be biased. The selection of a node is directly proportional to its reward potential if β equals unity. If β is larger than unity, then the selection of larger nodes will be favoured in the random selection process. On the other hand, if β is less than unity (but greater than zero), then smaller nodes will stand a higher chance of being selected relative to their reward potential. It is up to the user to specify a value for β. As will be noted in the following chapter, it is advisable to re-run an analysis for different values of β.

Routine START2 is based on the following procedure:

1: Specify β.

2: Standardise the rewards of all members of MN as noted above.

3: Stack the standardised rewards as a continuum from 0 to 1. This will allocate each node a slot proportional to its standardised reward.

4: Randomly derive a number between 0 and 1, and assess which node on the continuum identifies with that number.

5: Let this node be the next member of MY, the next node to be visited.

6: Test for condition 5.4 explained in the previous section.

7: If the condition is satisfied, then implement the improvement and return to step 2. If the condition is violated, then reject the last entry. Identify those nodes that can still be visited. For these, proceed from step 2. If no more nodes can be visited without violating condition 5.4, then the starting solution has been identified.

5.4.1.3 START3: The Logically Defined Starting Solution

This method arrives at a starting solution by routing from the depot according to a procedure based on a number of logically defined steps. The procedure used is based on a search for the node which offers highest reward, while remaining within a reasonable penalty to get there. The index used is the ratio of P_{ij}, the penalty that must be accepted to travel from i to j, to t_{ij}, the time taken to travel from i to j added to the time required to collect the reward at j. Each step, k, in the procedure requires the calculation of this index from the node last selected to all nodes not already incorporated in the route as:

$$\text{find the Max. } \frac{t_{D_k j}}{P_{D_k j}} \quad \text{for all } j,\ j \in \underline{MN} \qquad 5.5$$

The first search is for that node which yields the highest ratio from the depot. Subsequent searches are for the node that yields the highest ratio from the node last added to the sequence. After the introduction of each new point, the maximum penalty limit, PMAX, is checked by testing the following condition:

$$\sum_{i=1}^{k} t_{O_i D_i} + t_{D_k d} \leq PMAX \qquad 5.6$$

If this condition is violated, then the last node added is temporarily dropped from consideration as a node in the solution, and the process is repeated. If the condition is violated for all nodes not presently incorporated in MY, but which are members of MN, then the starting solution has been identified.

5.4.2 The CROSS Routine

The function of routine CROSS is to check whether the route presently being considered crosses itself at any stage. A self crossing route, by definition, cannot be the optimal route, and all self crosses must therefore be identified and eliminated. An example of a self crossing path is shown in Figure 5.3a.

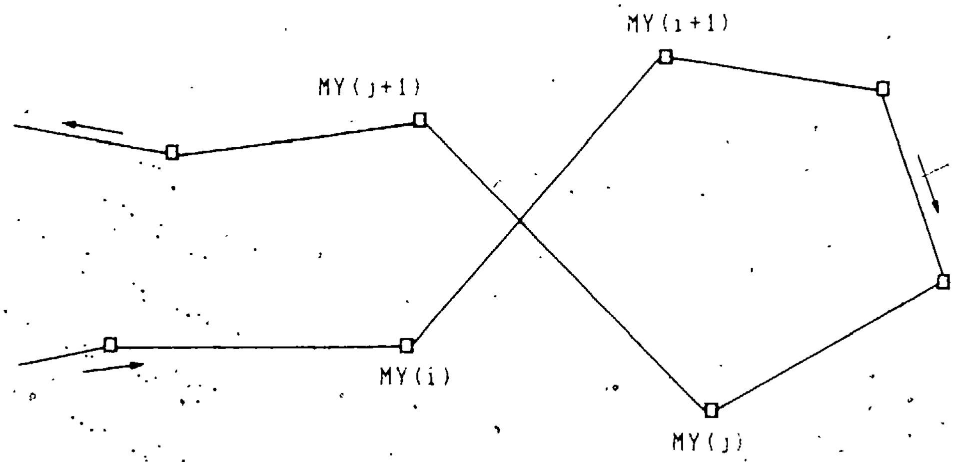
Routine CROSS detects the occurrence of a self crossing path by testing for the following condition:

$$\text{Test if } (P_{O_i O_j} + P_{D_i D_j}) < (P_{O_i D_i} + P_{O_j D_j}) \text{ for all } i,j \qquad 5.7$$

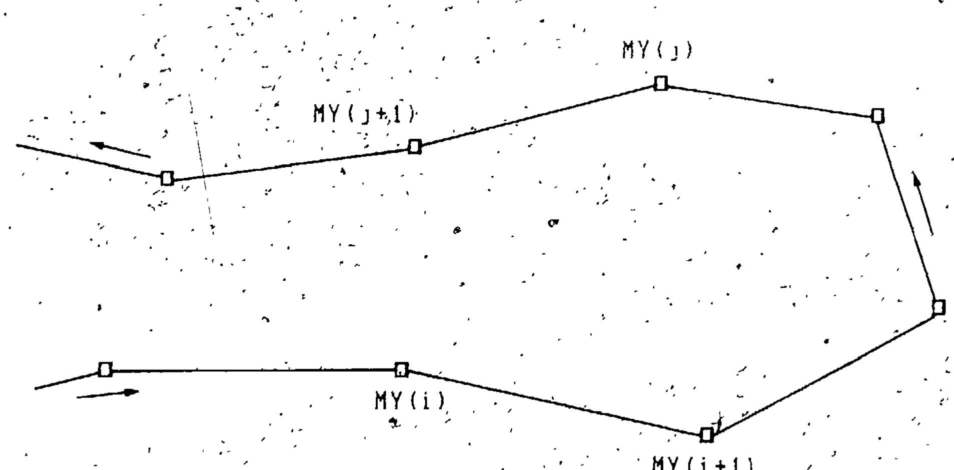
A self crossing path has been detected if this test holds true. Given such an occurrence, the route sequence vector MY needs to be modified so that MY_j becomes MY_{i+1}, and MY_{i+1} becomes MY_j. The direction of the route segment from MY_{i+2} to MY_{j-1} needs to be reversed so that

Figure 5.3 Example of a Self Crossing Path and its Elimination

a) A self crossing path is detected



b) The self crossing path is eliminated and the route sequence adjusted.



MY_{j-1} becomes MY_{i+2}, MY_{i+2} becomes MY_{j-1} ... . In other words, travel along part of the route sequence must now occur in the opposite direction. A graphical solution to a self crossing path is given in Figure 5.3b.

5.4.3 The SWITCH Routine

The function of routine SWITCH is to evaluate whether it is possible to improve the overall efficiency of the route presently under consideration. This is achieved by dropping a node within $\underline{MY}$ out of its present position, and substituting it elsewhere within the path. An example of such a swap is shown in Figure 5.4a.

Routine SWITCH evaluates for this type of improvement by testing for the following condition:

$$\text{Test if} \quad (P_{O_i D_i} + P_{O_{i+1} D_{i+1}} + P_{O_j D_j}) > (P_{O_i D_{i+1}} + P_{O_j D_i} + P_{O_{i+1} D_j}) \qquad 5.8$$

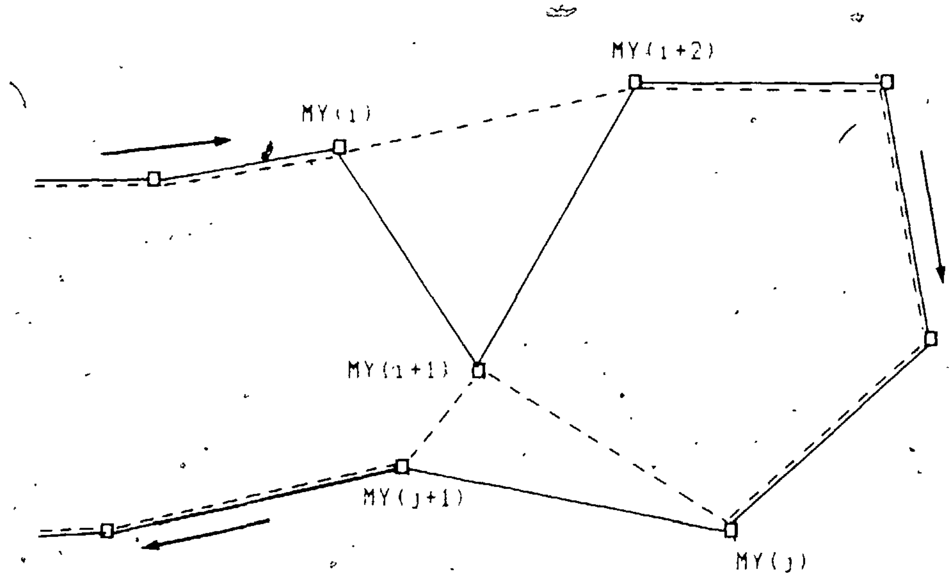
The route is not the most efficient if this test holds true. Swapping of nodes within the route is therefore implemented by taking MY_{i+1} from its present position within the route sequence vector $\underline{MY}$, and inserting it between MY_j and MY_{j+1}. No directional changes are involved in routine SWITCH; but the sequence of $\underline{MY}$ needs to be adjusted after every change. For a graphical representation of such a switch see Figure 5.4b.

5.4.4 Improvement Identification Routines

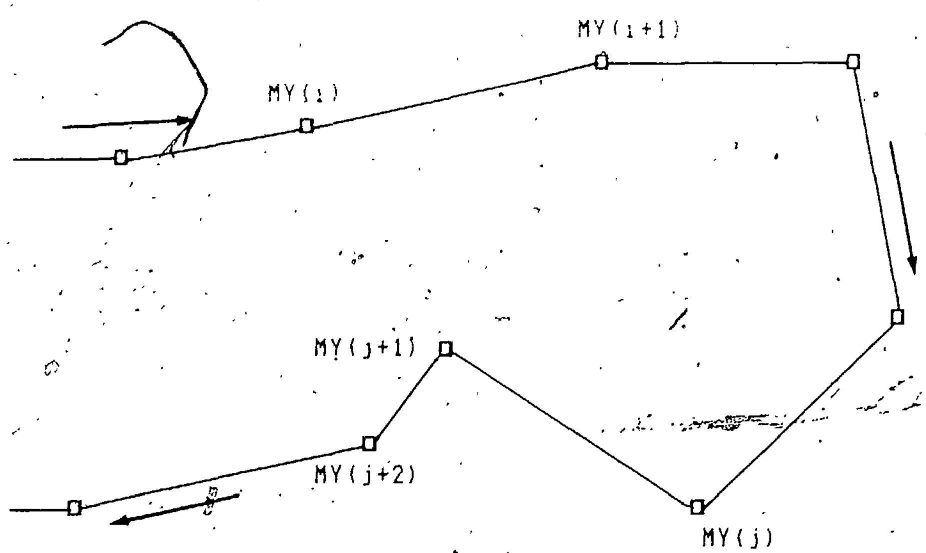
Improvements over a given route are identified by changes in the route sequence membership that increase overall reward collected, while

Figure 5.4 A Route Improvement Detected by Routine SWITCH

a) A possible improvement is detected (- - - - -)



b) The improvement is implemented and the route sequence adjusted



remaining within the limit of maximum acceptable penalty, FMAX. The processes that identify possible improvements over a given route apply the principles of node addition, substitution and subtraction. The simplest possible improvements are the ONE IN - ZERO OUT, ONE IN - ONE OUT and ONE IN - TWO OUT methods respectively. Higher level methods include TWO IN - ONE OUT, THREE IN - TWO OUT, TWO IN - TWO OUT and ONE IN - THREE OUT, or any other possible combination of node addition, substitution and subtraction.

The three simplest methods are computationally relatively uncomplicated and not very time consuming. Higher level methods become computationally increasingly complex since an X IN - Y OUT search will generally require $m^{Y} * (n-m)^{X}$ steps. A combination ot the three simplest improvement methods tend to detect most possible improvements for a given route, and they are therefore incorporated within the MVP heuristic. Two other improvement routines, DROP1 and DROP2, were designed to attempt to detect improvements not discovered by the above. These two routines operate by selectively forcing one or a number of nodes temporarily out of the solution, followed by a search for a route which yields higher return than the route that included the node(s) presently dropped. All five improvement routines are described in some detail below.

5.4.4.1 IMPROV1: The One IN - Zero Out Routine

The objective of this routine is to identify that node presently not incorporated in the route that would offer the largest possible increase in reward potential, and that can somehow be added into the existing route without its inclusion exceeding PMAX. The routine thus first finds:

$$\text{Maximum} \quad R_k \quad \text{for all } k,\ k \in \underline{MN} \qquad 5.9$$

The routine next tries to incorporate node k into the existing route at the minimum possible increase in penalty. This involves finding that link in the existing route that, if broken in order to be replaced by two new links connecting k into the route, minimises the overall penalty increase. Mathematically, find the minimum:

$$INC_{k_l} = P_{O_l k} + P_{k D_l} - P_{O_l D_l} \quad ; \quad O_l,\ D_l \in \underline{MY},\ l = 1, \ldots, m \qquad 5.10$$

INC_{k_l} stands for the increase in penalty on incorporating node l in the route after node k. For a graphical explanation see Figure 5.5a. Next, test for the following condition:

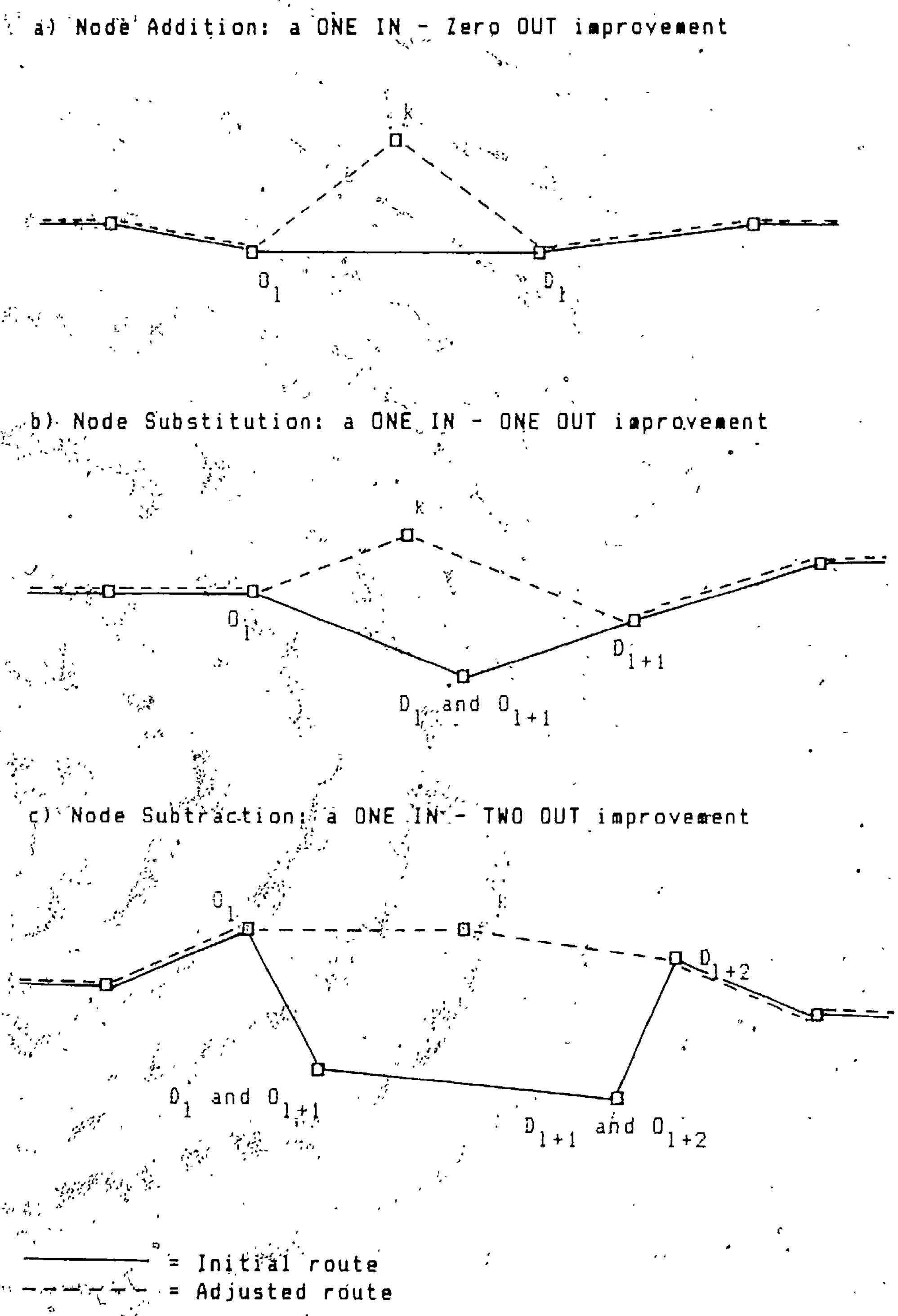
$$\text{Test if} \quad \hat{T} - t_{O_l D_l} + t_{O_l k} + t_{k D_l} \leq PMAX \qquad 5.11$$

$$\text{where} \quad \hat{T} = \sum_{i=1}^{m} t_{O_i D_i} \qquad 5.12$$

$\hat{T}$ stands for the total penalty before improvement.

If condition 5.11 is satisfied, then node k is incorporated into the route between O_l and D_l by adjusting vectors MY and MN. If condition 5.11 cannot be satisfied, then the routine returns and searches for the next largest R_k in vector MN, and proceeds as before. This routine is run again and again until there exist no more members in MN that can be incorporated somewhere within MY without violating condition 5.11.

Figure 5.5 Examples of Node Addition, Substitution and Subtraction



2 OF / DE 2

An alternative approach for this routine would have been to systematically search vector <u>MN</u>, and to incorporate into the existing route immediatly any member of <u>MN</u> that satisfies condition 5.11 subject to equation 5.10.

5.4.4.2 IMPROV2: The One In - One Out Routine

The objective of this routine is to take every member of vector <u>MN</u> systematically, and to consider substituting it for every member in <u>MY</u> (excluding the depot). A substitution is considered feasible if the reward potential of the member of vector <u>MN</u> is larger than that of the member of vector <u>MY</u> that is to be substituted. The idea is to search through all feasible substitutions for that one which offers the largest increase in reward. The substitution must remain within the maximum penalty threshold, PMAX. Mathematically this can be expressed as follows: Find the maximum

$$INC_k = R_k - R_{D_i} \quad ; \quad k \in \underline{MN},\ D_i \in \underline{MY},\ D_i \neq m \qquad 5.13$$

$$\hat{T} - (t_{O_i D_i} + t_{O_{i+1} D_{i+1}}) + (t_{O_i k} + t_{k D_{i+1}}) \leq PMAX \qquad 5.14$$

Graphically, such a substitution is shown in Figure 5.5b. Again, the identification and implementation of the maximum possible improvement in reward is favoured over successively accepting all feasible substitutions encountered that satisfy condition 5.14. This routine is run repeatedly until no further substitutions are possible that increase overall reward without violating PMAX.

5.4.4.3 IMPROV3: The One In - Two Out Routine

Simply subtracting one node from the existing route can clearly not yield an increase in overall reward. The first level of node subtraction that can possibly offer an improvement over the existing route is therefore that of adding one node from MN in favour of subtracting two nodes from MY. To keep this routine computationally simple, it is assumed that the two nodes to be dropped from MY are adjacent to each other. This implies that every time this routine is called approximately n(N-m) steps must be evaluated. If the assumption of adjacency was not included, then the number of steps would increase to approximately $n(N-m)^2$. The objective of this routine is thus to identify all situations where a member from MN offers a larger reward than the sum of the reward potentials of two adjacent members presently in MY (excluding the depot). Again, the routine identifies the situation that offers the largest reward increase possible within the limitations posed by PMAX. Mathematically, this can be expressed as follows: Find the maximum

$$INC_k = R_k - (R_{D_i} + R_{D_{i+1}})\ ;\ k \in \underline{MN},\ D_i \neq 1,\ D_i \neq m \qquad 5.15$$

subject to:

$$\hat{T} - (t_{O_i D_i} + t_{O_{i+1} D_{i+1}} + t_{O_{i+2} D_{i+2}}) + t_{O_i k} + t_{k D_{i+2}} \leq PMAX \qquad 5.16$$

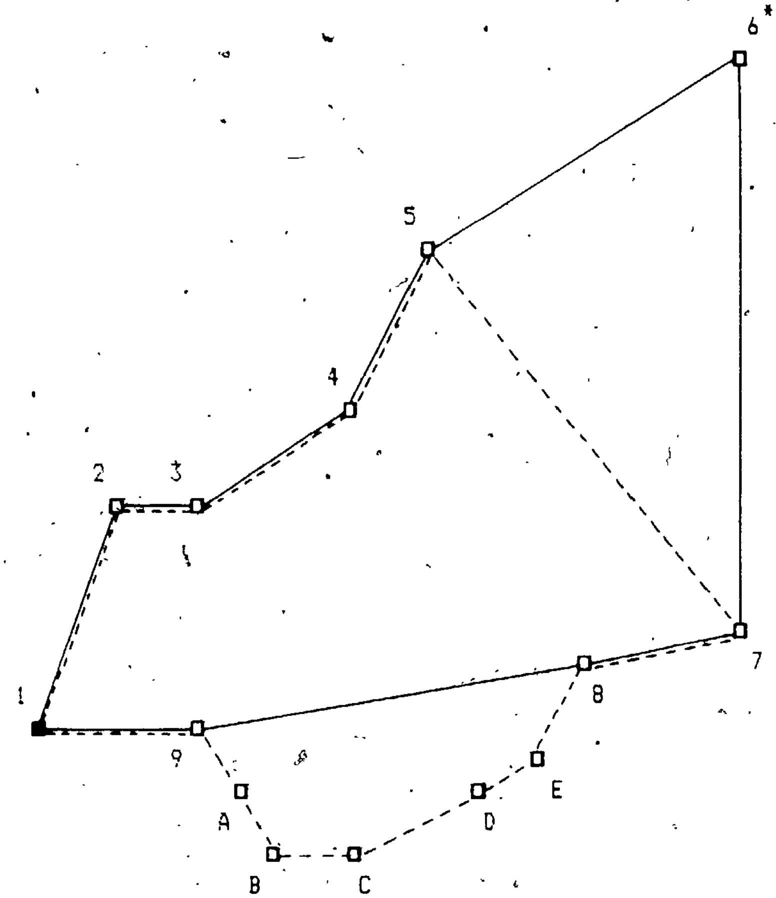
A graphical example of such an improvement is shown in Figure 5.5c. This routine, too, operates by identifying and implementing the maximum possible improvement in preference to implementing every possible detected improvement. Again, this routine is run until no further improvements can be detected within PMAX.

5.4.4.4 DROP1: The One Member Drop

Routine DROP1 was designed to identify the following scenario: Consider the example in Figure 5.6. For a given value of PMAX, the route shown as a solid line was identified. Node 6 entered the route at a very early stage because it offers a disproportionally large reward potential. There exists however a cluster of nodes to the south of the present route, nodes A through E, whose sum of rewards exceeds that of node 6. This cluster of nodes could be incorporated into the route without violating PMAX, should node 6 be dropped. Neither a ONE IN - ZERO OUT, a ONE IN - ONE OUT or a ONE IN - TWO OUT routine will however allow node 6 to be dropped out of the route sequence. Subroutine DROP1 identifies this sort of a situation by successively temporarily dropping every node in turn out of the present route sequence, as well as out of the solution space. Thus every member of <u>MY</u> (with the exception of the depot) is temporarilly removed from <u>MY</u>, but is not placed in <u>MN</u>. Each time a member of <u>MY</u> is dropped this way, the route is re-evaluated by implementing IMPROV1, IMPROV2 and IMPROV3.

If an improvement is detected that yields a higher total reward potential than the route that included the member presently dropped, then the improvement is accepted as the new optimal solution. The node temporarily removed from <u>MY</u> is thereafter placed in <u>MN</u>. If no improvements can be detected, then the presently removed member is returned into the route in its old place, and the next member in the sequence is temporarily dropped. Figure 5.6 graphically demonstrates an improvement identified by routine DROP1.

Figure 5.6 A Route Improvement Detected by Routine DROP1



■ = the depot

□ = nodes and their identity

——— = the initial optimal route

□6* : Node 6 offers a disproportionally large reward. Forcing node 6 temporarily out of the route sequence will allow nodes A through E to be incorporated.

- - - - - = the new optimal route

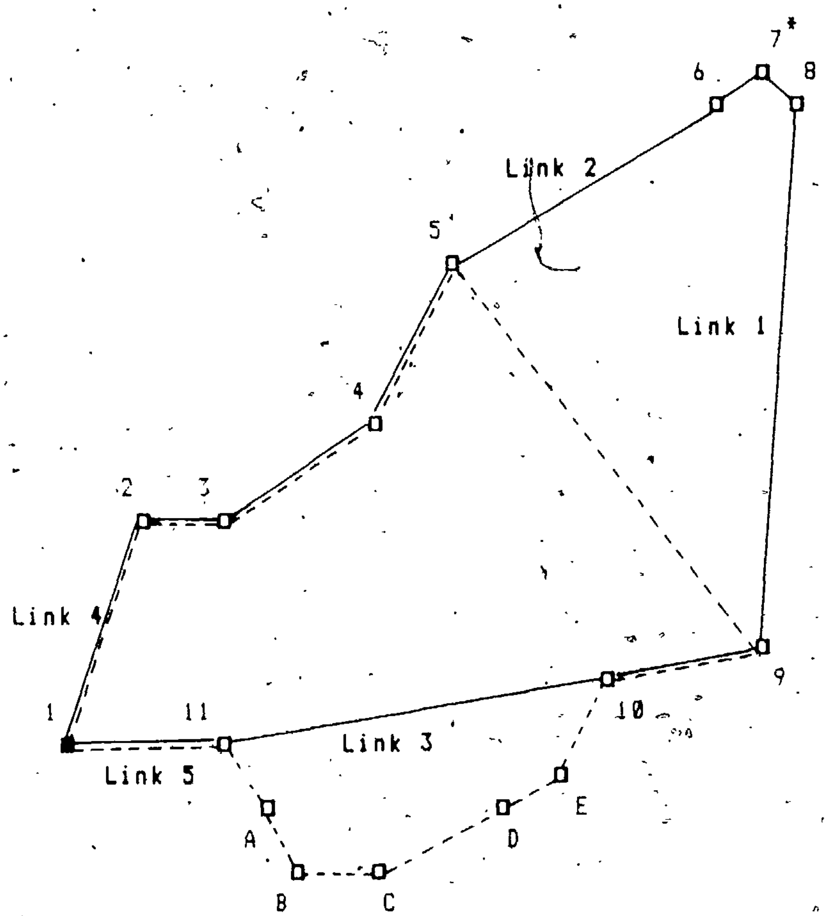
5.4.4.5 DROP2: Dropping Strings of Members

The objective underlying routine DROP2 is explained utilising the hypothetical example shown in Figure 5.7. As above, the optimal route identified for a given value of PMAX is shown as a solid line. This time, node 7 entered the route at a very early stage because it offers a disproportionally large reward potential. Its inclusion in the route ensured that adjacent nodes 6 and 8 became incorporated into the route. It is assumed that the reward potentials of nodes 6 and 8 are very small. The problem is that the sum of the rewards of the cluster of nodes A through E to the south still exceed the sum of the rewards of nodes 6,7 and 8. Should nodes 6,7 and 8 be dropped from the route, then nodes A through E could be incorporated into the route without violating PMAX, increasing overall reward. None of the improvement routines discussed so far detect this situation. An additional search procedure is therefore required.

The objective of routine DROP2 is to function as a search procedure that attempts to identify, and to temporarily remove from the problem space clusters of nodes that are far removed from the rest of the nodes within the solution, as for example the cluster of nodes 6,7 and 8 in Figure 5.7.

Routine DROP2 operates as follows: It starts by finding the link associated with the highest penalty, called LINK1. It subsequently finds the next highest links before and after LINK1, LINK2 and LINK3 respectively. Similarly, it identifies the next largest link before and after LINK2 and LINK3, LINK4 and LINK5 respectively... . The routine repeats this procedure until the largest links before and after the

Figure 5.7 A Route Improvement Detected by Routine DROP2



■ = the depot

□ = nodes and their identity

——— = the initial optimal route

□7* : Node 7 offers a disproportionally large reward, and will most likely ensure the inclusion of nodes 6 and 8 in the route sequence. Removing this remote cluster of nodes temporarily from the solution will allow nodes A through E to be incorporated.

------ = the new optimal route

previous largest links connect into the depot. This occurs in the hypothetical example shown in Figure 5.7 for LINK4 and LINK5.

Based on the assumption that all nodes lying between LINK2 and LINK1 form a cluster of nodes far removed from the rest of the solution, the procedure continues by temporarily removing that cluster of nodes from the problem space. This is achieved by linking the origin node of LINK2 with the terminal node of LINK1, and then implementing the above described improvement search.

If an improvement can be identified, such as that in the example given, then the new route is accepted as the optimal route, and the members of the temporarily removed string are added to MN. If no improvement can be identified, then the nodes between LINK1 and LINK3 are next temporarily removed, and the procedure is repeated as before. Ultimately, this routine will temporarily drop all nodes of the last identified optimal route. Given a problem with a very large number of nodes constrained by a relatively small value of PMAX, this situation may result in the identification of a completely new route that would previously not have been identified, but that may offer a larger overall reward. A graphical example of the identification of an improvement by routine DROP2 is shown in Figure 5.7.

CHAPTER SIX

6.0 Evaluation of the Performance of the MVP Heuristic

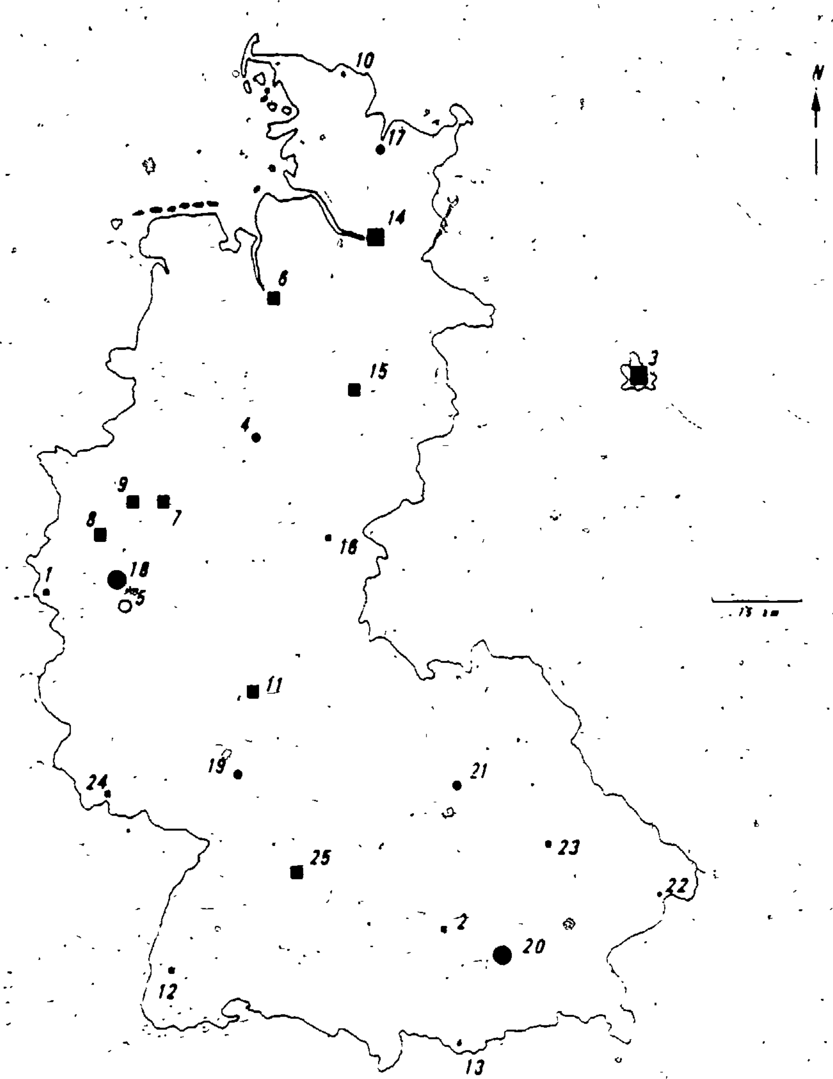
6.1 Introduction

To evaluate the performance of the MVP heuristic, the MVP program has been applied to a set of 25 cities located in West Germany, as shown in Figure 6.1. Bonn, the capital of West Germany, is treated as a depot and terminal node. The research objective is to identify the trade off between collecting reward potential offered at each of the 25 cities, and the penalty that must be accepted to travel to and service them. The problem therefore is to solve for the optimal subset of cities, and the optimal associated route sequence, to cover as much reward as possible for different maximum penalty constraints.

The populations of the cities, shown in Table 6.1, are treated as surrogates of reward that can be collected if the cities are visited. Intercity distances are treated as surrogates of penalty for travelling between the cities. The intercity distance matrix used for the analysis is that published on one of West Germany's road maps (Deutsche Centrale fur Tourismus). It is shown in Table 6.1. Distances represent the fastest, not necessarily the shortest distances between city pairs (Deutsche Centrale fur Tourismus). Given the absence of speed restrictions on West Germany's 'Autobahns', distances are therefore predominantly those for the shortest routes between two cities via the nearest 'Autobahn'.

The chapter commences by deriving and discussing the noninferior solution set, assuming that reward can be collected instantaneously upon arrival at a node. Subsequent analysis assumes that a penalty has to be

Figure 6.1 The 25 West German Cities



For identification of nodes refer to Table 6.1.

o = Depot

Size of Population (/1000)

· =	0 - 99	• =	100 - 249	• =	250 - 499
■ =	500 - 999	● =	1000 - 1500	■ =	> 1500

Table 6.1 Intercity Distance Matrix and Population Figures for the 25 West German Cities

Distance Matrix[@]

	1 Aachen	2 Augsburg	3 Berlin(West)	4 Bielefeld	5 Bonn	6 Bremen	7 Dortmund	8 Dusseldorf	9 Essen	10 Flensburg	11 Frankfurt	12 Freiburg	13 Garmisch	14 Hamburg	15 Hannover	16 Kassel	17 Kiel	18 Koln	19 Mannheim	20 Munchen	21 Nurnberg	22 Passau	23 Regensburg	24 Saarbrucken
1																								
2	587																							
3	660	586																						
4	287	557	394																					
5	90	512	601	221																				
6	306	683	393	153	308																			
7	166	607	492	112	116	251																		
8	75	571	560	180	78	316	60																	
9	114	563	530	152	102	291	36	33																
10	681	906	457	430	645	236	530	586	570															
11	250	360	550	270	172	475	267	230	250	670														
12	515	295	827	542	420	747	523	466	512	942	270													
13	710	119	680	670	631	860	726	661	717	1000	470	340												
14	524	719	290	271	470	119	363	428	460	167	503	775	903											
15	367	578	285	114	321	113	212	277	252	323	351	626	740	156										
16	347	434	381	123	266	284	175	236	210	477	201	473	566	310	165									
17	630	825	360	377	584	225	460	534	560	83	660	881	869	115	262	416								
18	80	536	572	190	26	329	85	46	74	660	190	446	640	441	290	250	547							
19	318	286	636	340	234	563	329	292	318	740	82	201	404	570	434	270	686	252						
20	634	66	582	619	571	770	606	629	666	963	410	354	90	795	650	496	901	509	344					
21	466	130	441	440	360	600	475	430	464	783	215	363	257	620	481	326	735	366	250	165				
22	804	251	663	674	606	825	700	663	686	1010	446	534	287	861	706	561	967	623	403	180	225			
23	573	134	532	563	404	704	579	542	500	887	319	420	220	730	506	430	836	507	362	122	104	121		
24	280	377	754	461	254	666	364	317	343	861	187	200	500	694	540	394	800	262	128	430	380	611	400	
25	454	161	633	524	347	660	442	406	431	860	196	204	290	802	547	302	790	365	120	220	182	400	207	230

ID	Name	Pop.[#]
1	Aachen	242
2	Augsburg	248
3	Berlin(West)	1967
4	Bielefeld	315
5	Bonn	284
6	Bremen	571
7	Dortmund	628
8	Dusseldorf	659
9	Essen	674
10	Flensburg	50
11	Frankfurt	631
12	Freiburg	174
13	Garmisch	30
14	Hamburg	1700
15	Hannover	550
16	Kassel	204
17	Kiel	261
18	Koln	1011
19	Mannheim	312
20	Munchen	1312
21	Nurnberg	496
22	Passau	40
23	Regensburg	131
24	Saarbrucken	205
25	Stuttgart	595

@ : Distances given are for the fastest, not necessarily the shortest routes in kilometers

\# : Population (000)

Source: Deutsche Centrale fur Tourismus

accepted not only for travelling the links, but also for collecting reward. The penalty for collecting reward is argued to be a linear function, proportional to some constant, Q, times the size of the reward. Four additional noninferior solution sets were derived for four different values of Q, and combined to yield a representation of the noninferior solution surface.

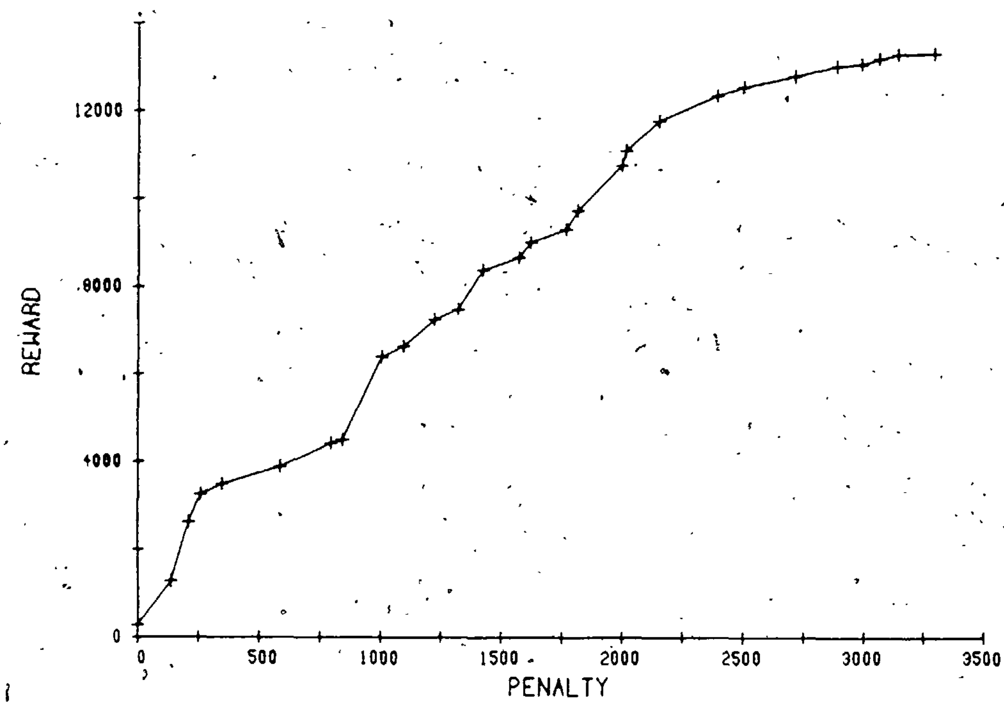
The analyses performed to derive the four noninferior solution sets discussed above were re-run a number of times. This procedure, as will be shown further on, allows for an evaluation of the overall accuracy of the heuristic, of the relative merits of the different procedures for deriving a starting solution, and of the accuracy of the heuristic for different sections of the noninferior solution curves. Examples of a number of inferior solutions encountered when running the heuristic are shown and discussed.

6.2 The Noninferior Solution Set

Figure 6.2 shows a 28 point approximation of the noninferior solution set obtained when assuming that reward at the different nodes can be collected instantaneously upon arrival. Table 6.2 gives the total reward collected, the total penalty accumulated and the exact nature of the routes for each of the 28 points.

Figure 6.2 shows that the noninferior solution curve is actually comprised of a number of discrete stepwise increases identified by a number of discrete points. The different steps can be interpreted to represent different clusters of nodes as demonstrated by a hypothetical example shown in Figure 6.3. Figure 6.3a shows two clusters of nodes, cluster A and cluster B. For simplicity's sake it is assumed that each node has the same reward potential. Stepwise increasing the maximum

Figure 6.2 A 28 Point Approximation of the Noninferior Solution Set for the 25 Node West Germany Problem



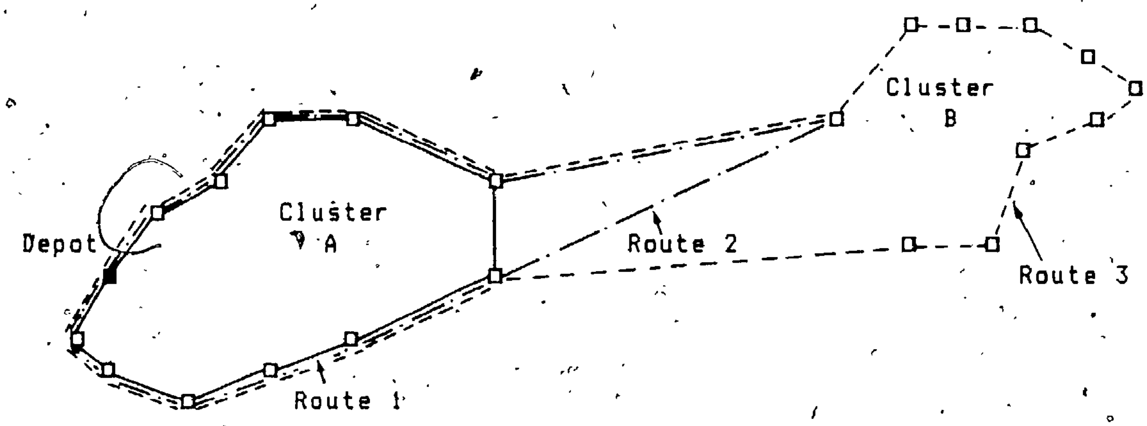
+ = Members of the noninferior solution set identified

Table 6.2 **Routes Identifying the Noninferior Solution Set**

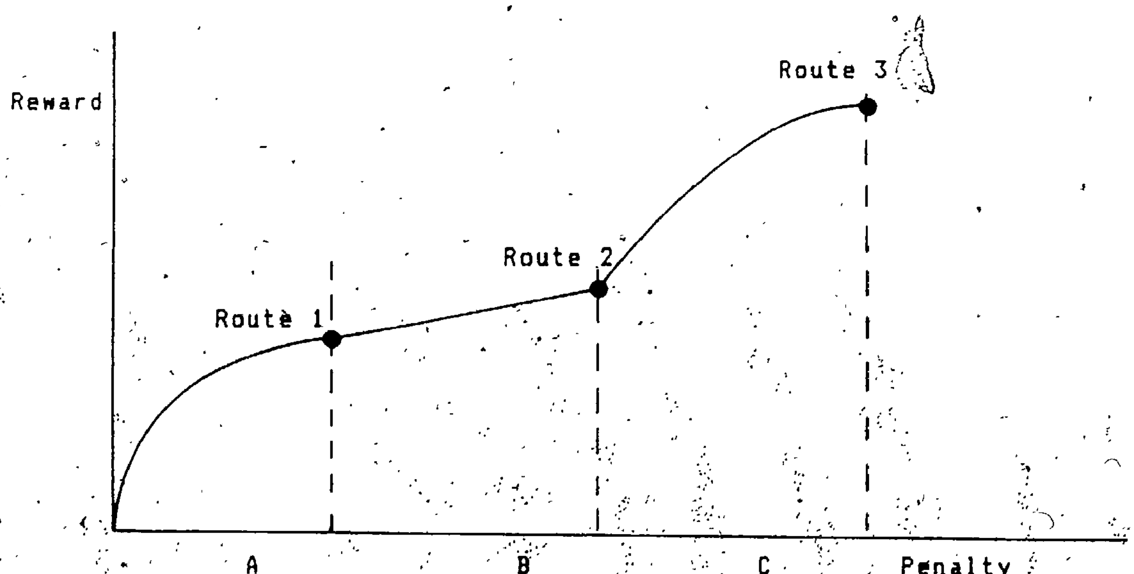
Route ID	Total Penalty	Total Reward	Route Sequence Identity
1	0	284	5
2	138	1295	5 18 5
3	211	2628	5 18 8 9 5
4	258	3256	5 18 7 9 8 5
5	347	3498	5 18 7 9 8 1 5
6	584	3887	5 18 8 9 7 11 5
7	795	4444	5 18 1 8 9 7 4 11 5
8	843	4514	5 18 8 9 7 4 11 19 5
9	1003	6400	5 18 4 15 14 6 7 9 8 5
10	1092	6642	5 18 4 15 14 6 7 9 8 1 5
11	1223	7235	5 18 8 9 7 4 15 6 14 16 11 5
12	1319	7273	5 18 1 8 9 7 4 6 14 15 11 5
13	1422	8052	5 18 8 9 7 6 14 3 15 5
14	1573	8652	5 18 8 9 7 4 6 14 3 16 11 5
15	1623	8998	5 18 8 9 7 4 15 6 14 3 11 5
16	1767	9310	5 18 8 9 7 4 15 6 14 3 11 19 5
17	1817	9736	5 18 1 8 9 7 4 15 6 14 3 21 11 5
18	2003	10763	5 18 8 9 7 15 14 3 21 20 2 25 11 5
19	2019	11099	5 18 8 9 7 4 6 14 3 21 20 2 25 11 5
20	2154	11780	5 18 8 9 7 4 15 6 14 3 21 23 20 2 25 11 5
21	2394	12334	5 18 1 8 9 7 4 15 6 14 3 21 23 20 2 25 11 19 5
22	2504	12539	5 18 1 8 9 7 4 15 6 14 3 21 23 20 2 25 24 19 11 5
23	2717	12800	5 18 1 8 9 7 4 15 6 14 17 3 21 23 20 2 25 24 19 11 5
24	2891	13004	5 18 1 8 9 7 4 16 15 6 14 17 3 21 23 20 2 25 24 19 11 5
25	2993	13054	5 18 1 8 9 7 4 16 15 6 10 17 14 3 21 23 20 2 25 24 19 11 5
26	3064	13178	5 18 1 8 9 7 4 16 15 6 14 17 3 21 23 20 2 25 12 24 19 11 5
27	3146	13268	5 18 1 8 9 7 4 16 15 6 10 17 14 3 21 23 22 20 2 25 12 24 19 11 5
28	3297	13298	5 18 1 8 9 7 4 16 15 6 10 17 14 3 21 23 22 20 13 2 25 12 24 19 11 5

Figure 6.3 A Hypothetical Example to Demonstrate the Occurrence of Steps in an MVP Type Noninferior Solution Set

a) Hypothetical example of the locations of a set of potential demand nodes



b) The resultant noninferior solution set

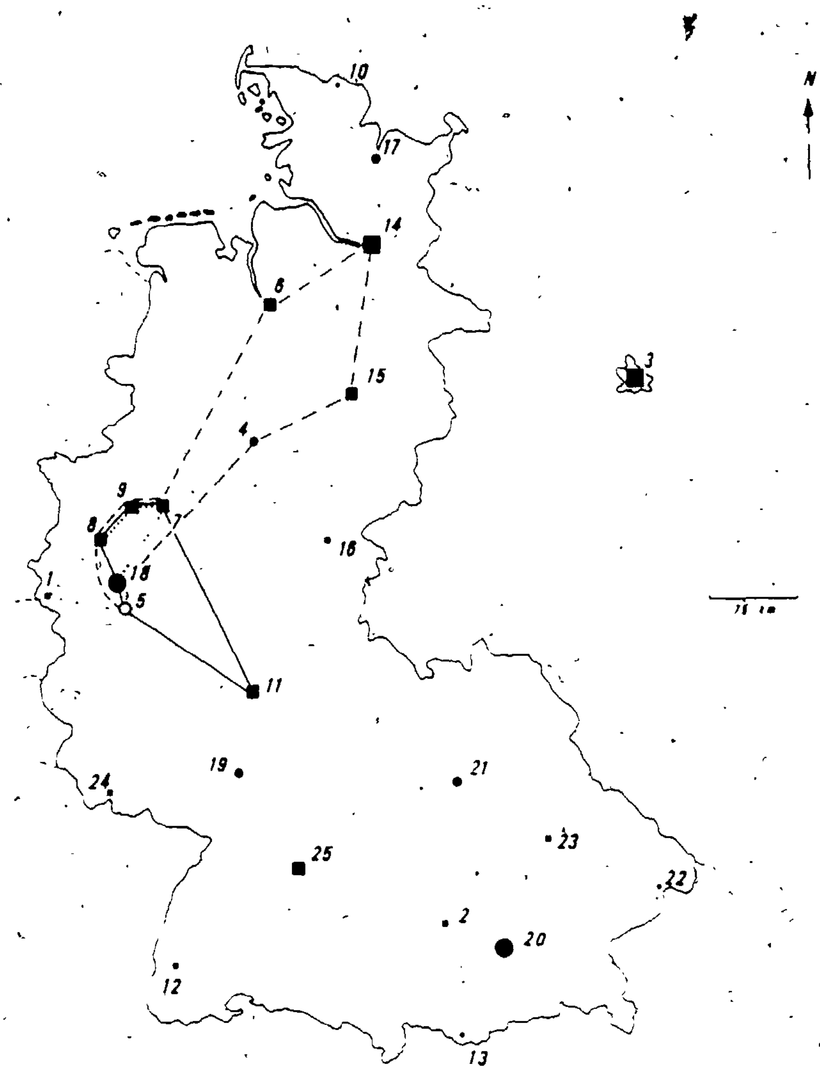


penalty constraint will initially result in the route successively adding more and more nodes from cluster A into the solution, resulting in the relatively uniform growth of part A of the resultant noninferior solution curve. Once all the nodes of cluster A are incorporated into the route, a large increase in penalty is required to allow the inclusion of one further node, the first node of cluster B in Figure 6.3a. This results in the relatively flat portion of part B of the noninferior solution curve. Once the first node of cluster B is reached, subsequent nodes can again be incorporated into the route at relatively little increase in penalty, resulting in part C of Figure 6.3a, the second steeper growth of the noninferior solution curve.

To demonstrate the growth of the route when increasing the maximum penalty constraint for the West German problem, and to demonstrate where the different step breaks occur, the routes represented by eight of the points on the noninferior solution set, shown in Figure 6.2, are examined in Figures 6.4 to 6.7.

Point 5 on the noninferior solution set is shown in Figure 6.4. as route 5. This route is an example of a route that covers nodes in close proximity to the urban concentration of West Germany's Ruhr Valley. Route 6 in Figure 6.4 is the route represented by point 6 on the noninferior solution curve. It is the first route encountered that breaks from the Ruhr Valley cluster to include the urban industrial concentration around the Frankfurt region. The next route shown, route 9 represents point 9 on the noninferior solution curve. The maximum penalty has by now been sufficiently increased to allow the incorporation of the large North Sea ports of Hamburg and Bremen into

Figure 6.4 Noninferior Routes 5, 6 and 9



.... =	Route 5 :	5 18 7 9 8 1 5
——— =	Route 6 :	5 18 8 9 7 11 5
- - - =	Route 9 :	5 18 4 15 14 6 7 9 8 5

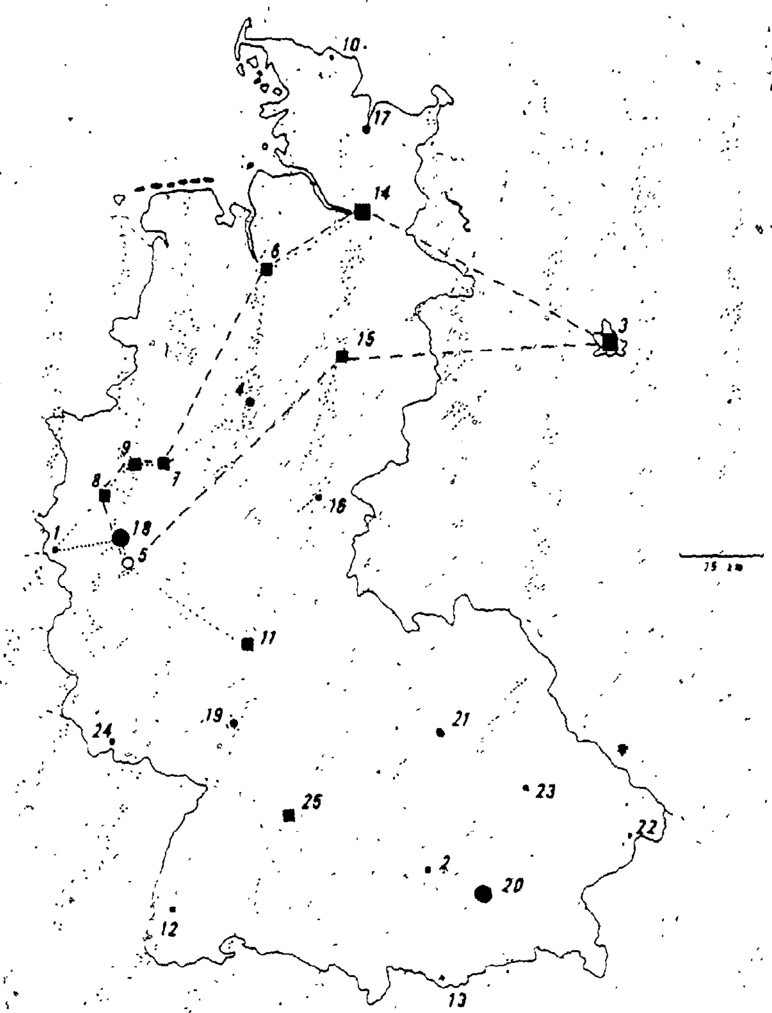
For identification of nodes refer to Table 6.1.

o = Depot

Size of Population (/1000)

· =	0 - 99	• =	100 - 249	• =	250 - 499
■ =	500 - 999	● =	1000 - 1500	■ =	> 1500

Figure 6.5 Noninferior Routes 12 and 13



┈┈┈ = Route 12 : 5 18 1 8 9 7 4 6 14 15 11 5

– – – = Route 13 : 5 18 8 9 7 6 14 3 15 5

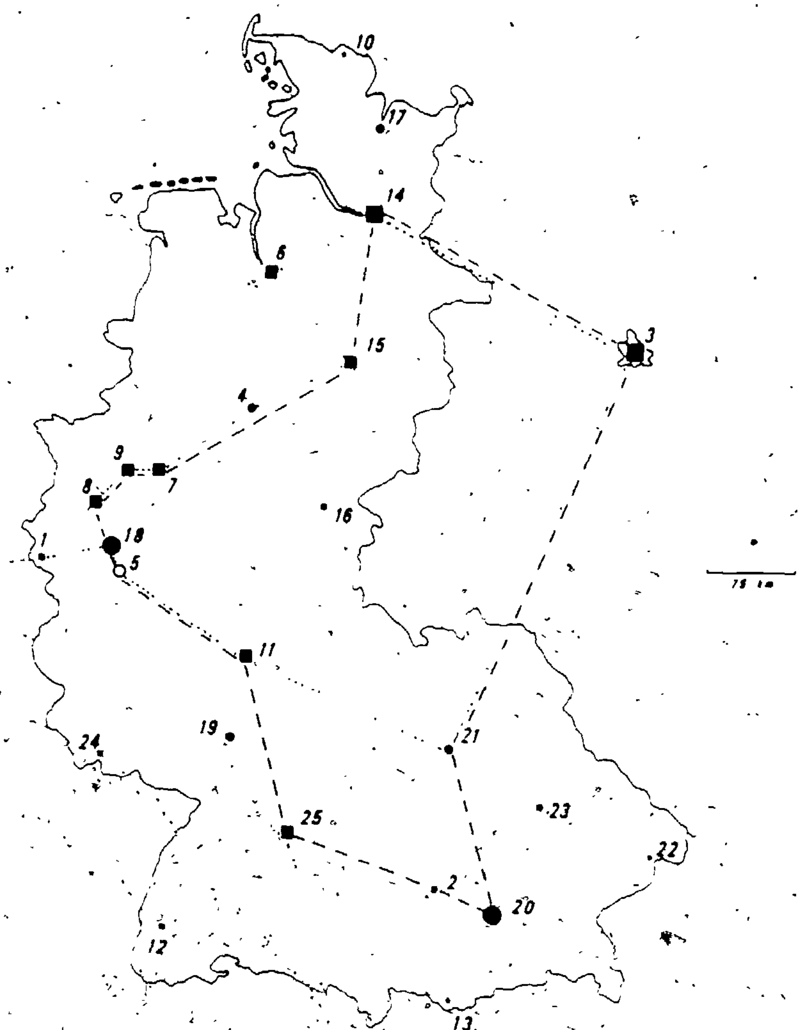
For identification of nodes refer to Table 6.1.

o = Depot

Size of Population (/1000)

·	=	0 - 99	•	=	100 - 249	•	=	250 - 499
■	=	500 - 999	●	=	1000 - 1500	■	=	> 1500

Figure 6.6 Noninferior Routes 17 and 18



..... = Route 17 : 5 18 1 8 9 7 4 15 6 14 3 21 11 5
– – – = Route 18 : 5 18 8 9 7 15 14 3 21 20 2 25 11 5

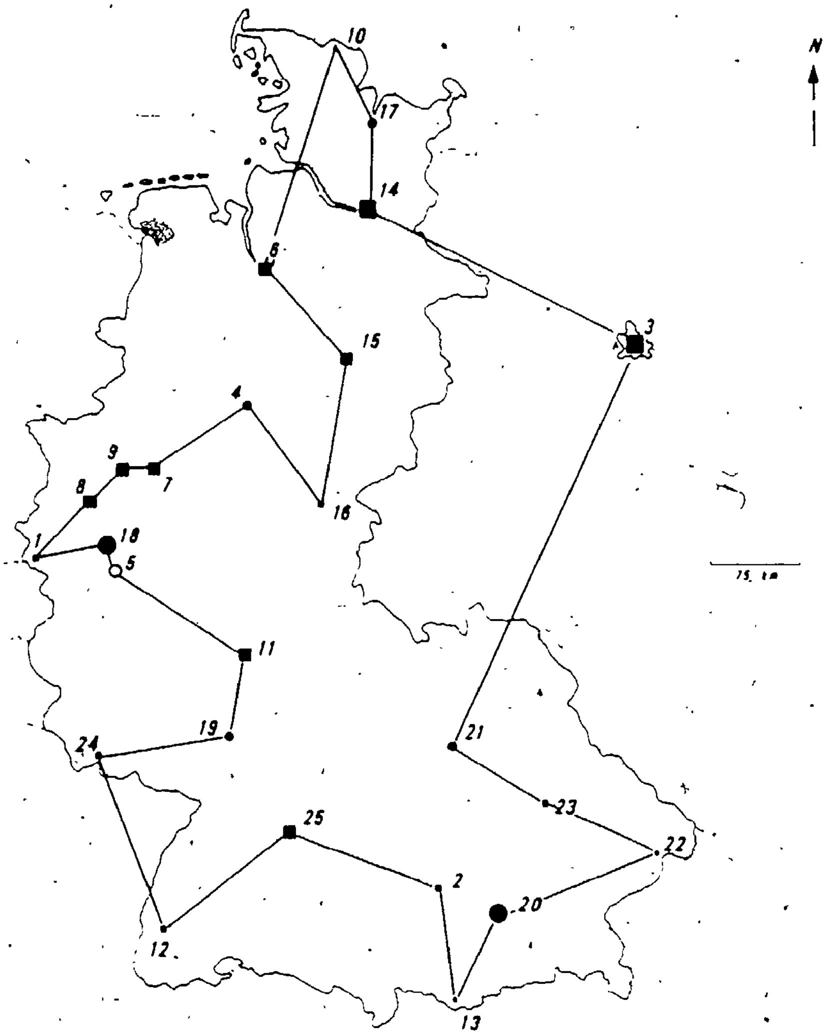
For identification of nodes refer to Table 6.1.

○ = Depot

Size of Population (/1000)

• =	0 - 99	• =	100 - 249	• =	250 - 499
■ =	500 - 999	● =	1000 - 1500	■ =	> 1500

Figure 6.7 Noninferior Route 28



——— = Route 28 : 5 18 1 8 9 7 4 16 15 6 10 17 14 3 21 23 22 20 13 2 25 12 24 19 11 5

For identification of nodes refer to Table 6.1.

o = Depot

Size of Population (/1000)

·	= 0 - 99	▪	= 100 - 249	•	= 250 - 499
■	= 500 - 999	●	= 1000 - 1500	■	= > 1500

the route. To do so, the Frankfurt cluster had however to be dropped back out of the solution.

Points 12 and 13 on the noninferior solution set are represented by routes 12 and 13 in Figure 6.5. The steep growth in reward between these two routes, evident from Figure 6.2, is due to the fact that route 13 for the first time allows the inclusion of the rather large but isolated city of West Berlin. Points 17 and 18 on the noninferior solution set, represented by routes 17 and 18 in Figure 6.6, show the nature of a route prior to and following the inclusion of the large city of Munich. Figure 6.7 shows the route which underlies point 28 on the noninferior solution curve, the route that connects all 25 cities. This route represents the solution to the general travelling salesman problem of connecting all the 25 cities at least penalty possible.

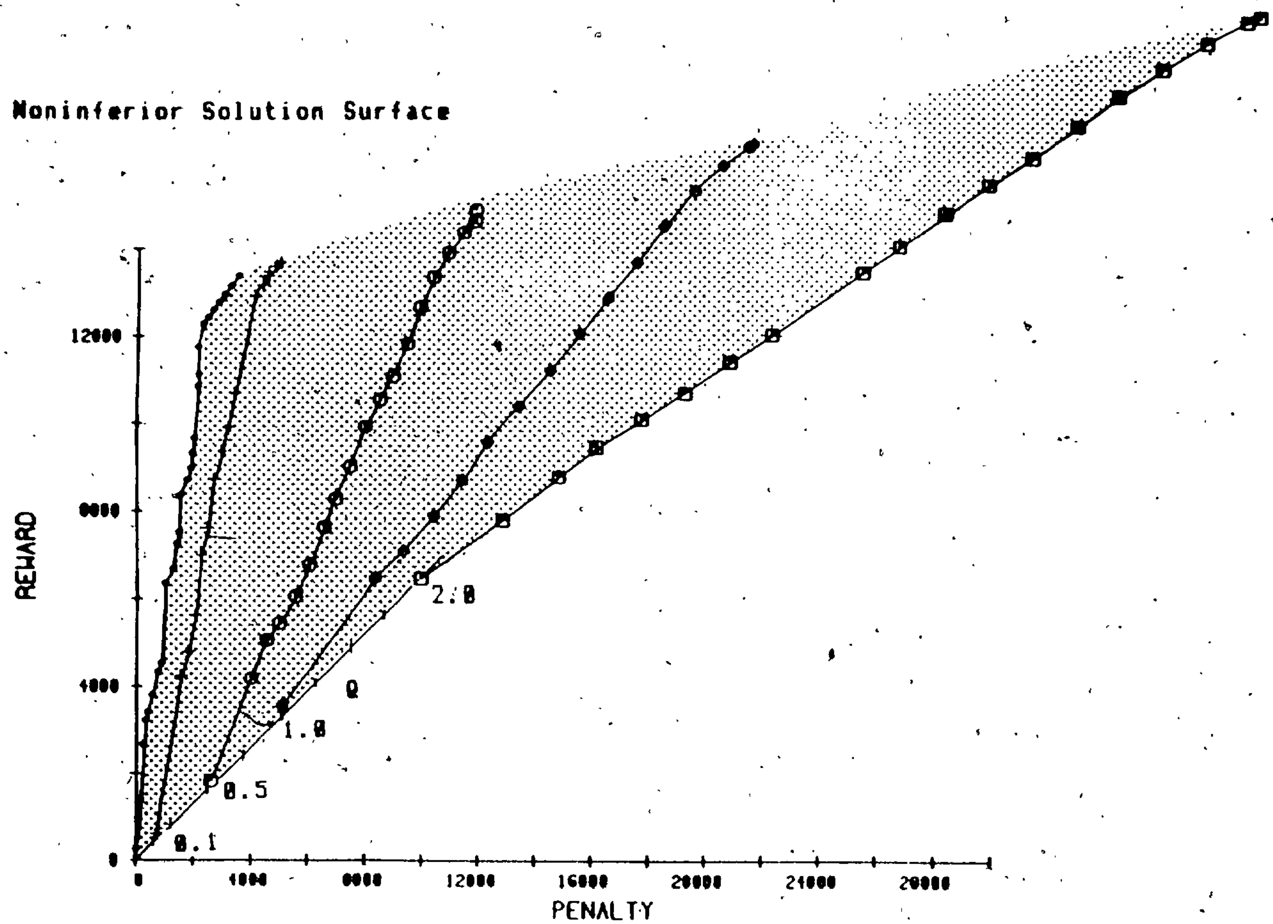
6.3 The Noninferior Solution Surface

As discussed earlier, the assumption underlying a noninferior solution surface is that some penalty must also be accepted for collecting reward at the different nodes, and that the penalty for servicing one unit of reward, Q, may vary. A third dimension can therefore be added to the problem space, one that allows an evaluation of how the noninferior solution set changes for changing values of Q.

To obtain a representation of this noninferior solution surface for the West Germany problem, noninferior solution sets are derived for four different values of Q, Q = 0.1, 0.5, 1.0 and 2.0 respectively. Each of these four noninferior solution curves has been approximated by 16 points. The four curves and the resultant noninferior solution surface are shown in Figure 6.8. It becomes evident from this figure that the

Figure 6.0

The Noninferior Solution Surface



+ = noninferior solution set for Q = 0.1
o = noninferior solution set for Q = 0.5
◇ = noninferior solution set for Q = 1.0
□ = noninferior solution set for Q = 2.0
(shaded) = approximation of the noninferior solution surface

overall slopes of the noninferior solution curves decrease in steepness and that the curves smooth out for increasing values of Q.

The decreasing steepness of the slopes for increasing values of Q is intuitively obvious; for a given route sequence and Q = 0, let total reward, $\Sigma R_i = R$, and total penalty, $\Sigma\Sigma P_{ij} = P$. If, for the same route sequence, Q was now increased so that Q > 0, then total reward will remain the same. Total penalty will however increase by Q*R, since P now equals $\Sigma\Sigma P_{ij} + \Sigma Q*R_i$. The smoothing out of the curves for increasing values of Q can be explained two ways. First, the four curves are approximated by only 16 points compared to 28 points for the noninferior solution set shown in Figure 6.2. The fewer the number of points to approximate a noninferior solution curve, the smoother that curve will be. Second, since the penalty for collecting one unit of reward increases with an increasing value of Q, total penalty for collecting all the reward at any one city will increase, thereby decreasing the slope and smoothing the curve.

6.4 Performance of the Heuristic

To allow for an evaluation of the performance of the MVP heuristic, each of the 16 points, approximating each of the four noninferior solution sets, was derived by solving for an optimal solution feasible within a specified maximum penalty constraint 16 times. This resulted in 16*16*4 or 1024 runs of the MVP heuristic.

Sixteen runs to evaluate for each point were necessary to test the performance of the heuristic utilising different techniques to generate a starting solution. To do so, the sixteen runs for each point were subdivided as follows: One run utilised the logically defined starting procedure discussed in section 5.4.1.3, routine START3. The other fifteen runs utilised the random starting procedure discussed in section 5.4.1.2, routine START2. These fifteen runs were divided into three groups of five runs each. Each group utilised a different value for β in equation 5.4. The three different values for β were 0.1, 1.0 and 5.0. They respectively increase the relative probability of including smaller cities in the starting solution, give all cities probabilities of entering the starting solution equal to their reward potential, and increase the relative probability of including larger cities in the starting solution. The user defined starting solution outlined in section 5.4.1.1, routine START1, has not been included in the evaluation since this technique relies on the skills of the analyst. An evaluation of the performance of this starting procedure would thus simply evaluate the analyst's familiarity with the problem researched.

Of all the sixteen runs evaluated for every individually specified maximum penalty constraint, that run which yielded the largest possible reward was treated as the optimal solution, and represented a point on the noninferior solution curve. The results obtained for each of the sixteen runs were then evaluated against this optimal result. This yielded three indices, the first indicating how often each technique for generating a starting solution led the MVP heuristic to identify the optimal run found, and the second and third indices allowing for the identification of the worst solution encountered, and the average departure of inferior solutions from the optimal solution found.

To reduce the volume of output produced by the 1024 analyses, and to simplify the interpretation of the results, the findings have been aggregated into four groups or quarters. The four quarters represent those points approximating the first, second, third and fourth quarter of the noninferior solution set respectively. For the four different quarters for each of the four noninferior solution sets, Table 6.3 shows the following three values for each of the four different starting approaches evaluated: the probabilities that a given starting approach will lead to the optimal answer within any one of the given quarters, what percentage of the optimal answer was achieved by the most inferior solution encountered within the different quarters, and the average percentage of inferior answers to optimal answers within the different quarters. Table 6.4 summarises Table 6.3.

The two tables show that of all 1024 runs, the most inferior run encountered yields results within 78.7 percent of the associated optimal solution encountered. This is quite low. On average, inferior solutions did however yield results within 98.1 percent of the optimal solution. Table 6.4 shows that the probabilities of obtaining the optimal solution decrease marginally with increasing values of Q. Table 6.3 suggests that the heuristic performs marginally better in the first quarter of each noninferior solution set.

There appears to be no obvious advantage to giving preference to any one of the four starting techniques over the others. It was found that there is a 95 percent chance of arriving at the optimal solution found if evaluating for the optimal solution four times, utilising each of the four starting approaches once. To ensure good results, it is therefore suggested that any analysis utilising the MVP program should be run four times, once for each of the four starting techniques.

Table 6.3 **Detailed Results for the Evaluation of the Performance of the MVP Heuristic**

		Logically Defined Starting Technique			Random Starting Technique $\beta=0.1$			Random Starting Technique $\beta=1.0$			Random Starting Technique $\beta=5.0$		
Q	Quarters	I	II	III	I	II	III	I	II	III	I	II	III
0.1	1st	1.00	100.0	100.0	0.95	96.4	96.4	0.95	87.3	87.3	0.95	99.9	99.9
0.1	2nd	0.75	99.7	99.7	0.65	89.9	96.0	0.55	89.9	97.3	0.60	89.9	96.5
0.1	3rd	0.50	98.3	99.3	0.60	96.9	98.5	0.60	96.9	98.8	0.55	96.9	98.7
0.1	4th	0.75	98.8	98.8	0.95	99.6	99.6	0.90	97.8	97.8	1.00	100.0	100.0
0.5	1st	0.75	98.1	98.1	1.00	100.0	100.0	1.00	100.0	100.0	1.00	100.0	100.0
0.5	2nd	0.75	98.7	98.7	0.70	95.7	98.7	0.75	90.1	96.1	0.80	95.7	98.0
0.5	3rd	0.75	99.4	99.4	0.85	98.3	99.2	0.70	95.2	98.6	0.80	95.2	97.6
0.5	4th	1.00	100.0	100.0	0.98	98.8	99.4	0.80	98.8	98.8	0.95	98.8	98.8
1.0	1st	1.00	100.0	100.0	0.60	85.5	94.0	0.60	78.7	94.1	0.45	78.7	91.0
1.0	2nd	0.75	95.8	95.8	0.60	95.8	97.7	0.70	96.4	98.2	0.85	96.0	97.4
1.0	3rd	0.25	98.2	98.7	0.75	95.8	97.8	0.50	97.2	99.1	0.45	95.5	98.3
1.0	4th	0.75	99.3	99.3	0.95	99.6	99.6	0.85	99.1	99.3	0.80	99.1	99.4
2.0	1st	0.75	98.1	98.8	0.45	84.6	93.3	0.55	84.[illegible]	93.4	0.55	89.3	96.1
2.0	2nd	0.50	99.6	99.8	0.50	96.4	98.0	0.55	97.0	98.5	0.35	97.0	98.6
2.0	3rd	0.25	98.9	99.2	0.50	98.3	99.4	0.55	98.1	99.3	0.35	97.4	99.3
2.0	4th	0.25	98.9	99.5	0.35	97.2	99.3	0.30	97.2	99.3	0.55	97.2	99.4

I probability of obtaining the optimal answer
II percent reward obtained by the worst case
III percent reward obtained on average

Table 6.4 **Summarised Results for the Evaluation of the Performance of the MVP Heuristic**

	Logically Defined Starting Technique			Random Starting Technique $\beta=0.1$			Random Starting Technique $\beta=1.0$			Random Starting Technique $\beta=5.0$		
	I	II	III	I	II	III	I	II	III	I	II	III
Q = 0.1	0.75	98.3	99.5	0.78	89.9	97.6	0.75	87.3	95.3	0.78	89.9	98.8
Q = 0.5	0.81	98.1	99.1	0.86	95.7	99.3	0.81	90.1	98.4	0.89	95.2	98.6
Q = 1.0	0.68	98.2	98.5	0.72	88.5	97.3	0.66	78.7	97.7	0.63	78.7	96.5
Q = 2.0	0.44	98.9	99.3	0.45	84.6	97.5	0.49	84.6	97.6	0.45	89.3	98.4
Average	0.67	98.4	99.1	0.70	89.7	97.9	0.68	85.2	97.3	0.89	88.3	98.1

I probability of obtaining the optimal answer
II percent reward obtained by the worst case
III percent reward obtained on average

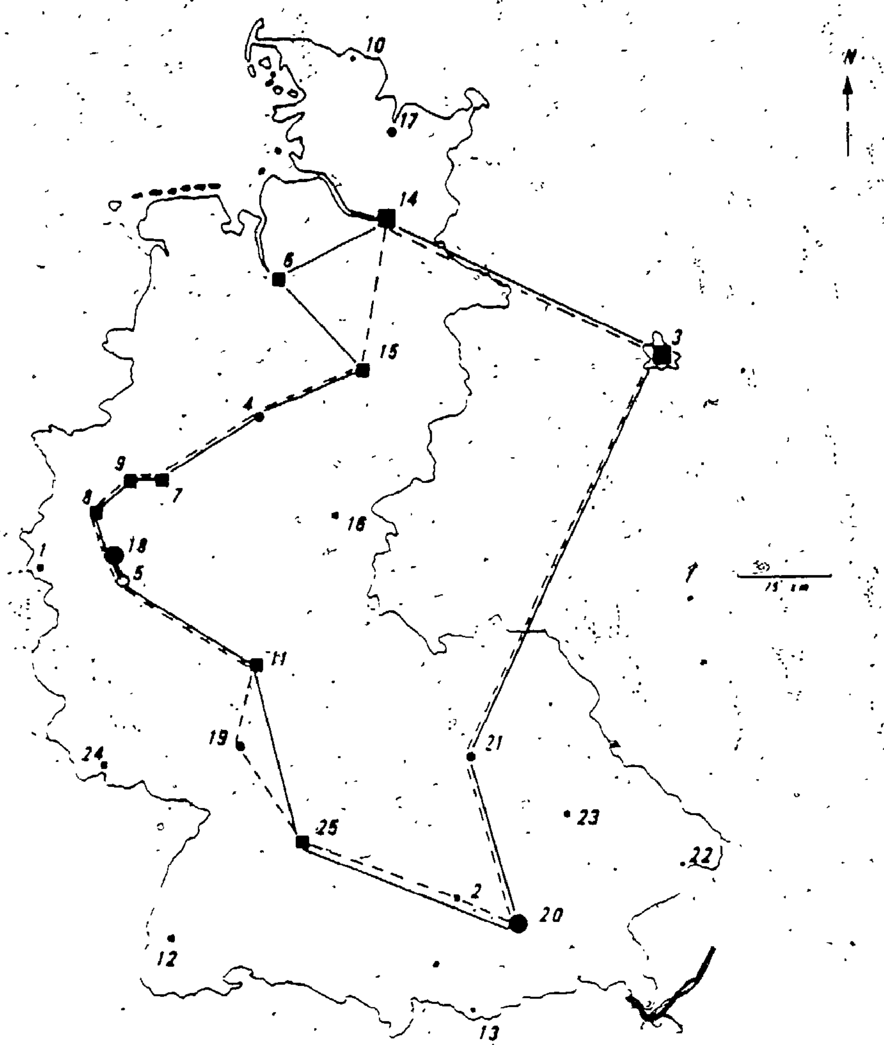
6.5 Examples of Inferior Solutions

Two cases which resulted in inferior solutions are now inspected in some detail. The examples selected are the following.

The first concerns the noninferior solution set for Q=2.0. Constraining the maximum acceptable penalty to not exceed 24950 units, the sixteen individual runs utilising the four different starting approaches identified two routes which are shown in Figure 6.9. The optimal route yields a total reward of 11401 units, the other 11390 units, a 0.1 per cent loss. To identify the optimal solution from the less optimal in one search would have required a ONE IN - TWO OUT swap. Such a swap is actually included in the MVP heuristic by routine IMPROV3. Routine IMPROV3 assumes however that the two nodes to be dropped out of the route are adjacent within the route sequence vector MY, a condition not satisfied here. As outlined in section 5.4.4.3, the inclusion of a ONE IN - TWO OUT swap not including the adjacency assumption is computationally very time consuming. In order to keep the heuristic manageable for larger problems, it was therefore not included in the MVP heuristic.

The second example concerns the noninferior solution set for Q= 0.1. Constraining the maximum acceptable penalty to not exceed 3032 units, the sixteen runs identified the three different routes shown in Figure 6.10. The optimal solution was identified only twice. This solution offers a total reward of 10334 units. The second and third most optimal solutions were identified seven times each and yield total rewards of 10331 and 10288 units respectively. The loss in rewards for the second and third most optimal routes relative to the most optimal are thus 0.01 and 0.45 per cent respectively. The second most optimal

Figure 6.9 Example One of an Inferior Solution Derived



—— = Optimal route sequence found. Reward = 11,401. Penalty = 24,888.
Sequence: 5 18 8 9 7 4 15 6 14 3 21 20 25 11 5

---- = Inferior sequence derived. Reward = 11,390. Penalty = 22,941.
Sequence: 5 18 8 9 7 4 15 14 3 21 20 2 25 19 11 5

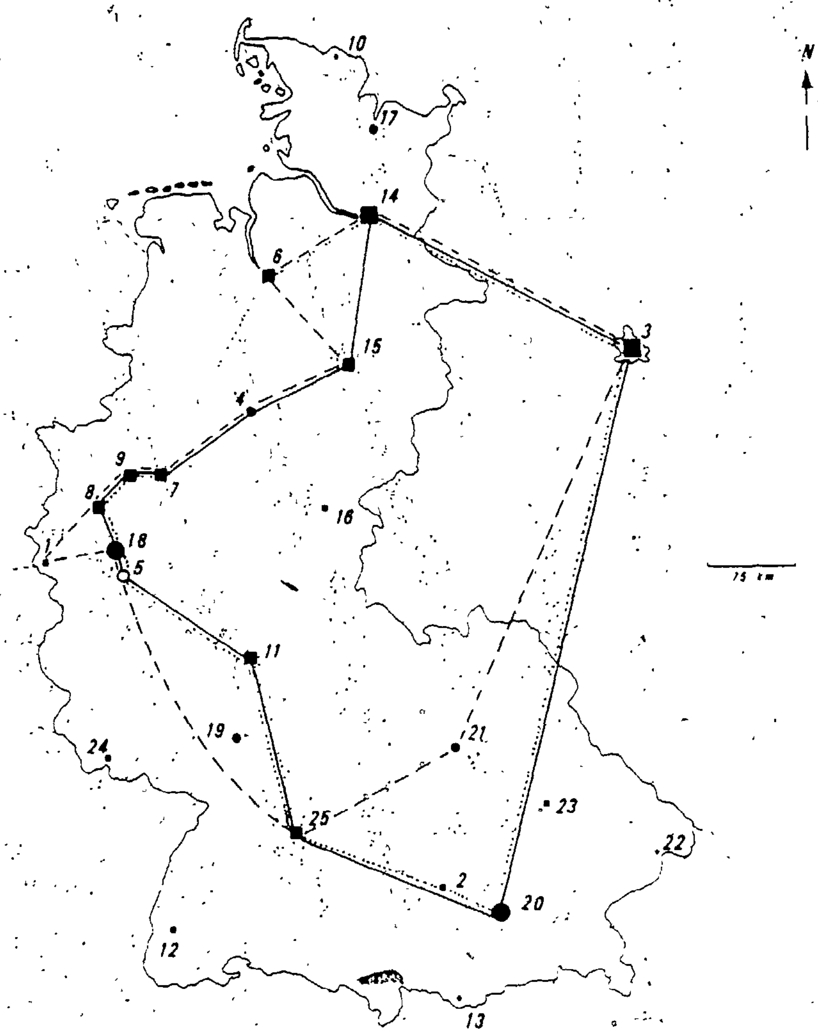
For identification of nodes refer to Table 6.1.

o = Depot

Size of Population (/1000)

· =	0 - 99	• =	100 - 249	• =	250 - 499
▪ =	500 - 999	● =	1000 - 1500	■ =	> 1500

Figure 6.10 Example Two of Inferior Solutions Derived



		Reward
—— = Optimal sequence :	5 18 8 9 7 4 15 14 3 20 25 11 5	: 10334
- - - = Inferior sequence:	5 18 1 8 9 7 4 15 6 14 3 21 25 5	: 10331
..... = Inferior sequence:	5 18 8 9 7 6 14 3 20 2 25 11 5	: 10228

For identification of nodes refer to Table 6.1.

o = Depot

Size of Population (/1000)

· =	0 - 99	• =	100 - 249	● =	250 - 499
■ =	500 - 999	● =	1000 - 1500	■ =	> 1500

route would have led to the identification of the most optimal route by one further ONE IN - THREE OUT substitution. This ONE IN - THREE OUT swap could not have assumed that the nodes to be dropped are adjacent. The third most optimal route differs from the most optimal route by two nodes. A TWO IN - TWO OUT substitution not including an assumption of node adjacency would therefore have led to the identification of the most optimal answer in one additional search here. Again a ONE IN - THREE OUT or a TWO IN - TWO OUT improvement search were however not incorporated in the MVP heuristic to keep it computationally simple and time efficient.

6.6 Conclusion

The two cases discussed above demonstrate the trade off between keeping the MVP heuristic computationally simple and time efficient, and producing the most optimal answer more frequently by including improvement searches based on higher levels of node addition, susbstitution and subtraction. Tables 6.3 and 6.4 do suggest that whereas the trade off results in a considerable loss of reward potential in a few isolated instances, inferior solutions obtained generally achieve results very close to the most optimal solutions found. The tables further suggest that by re-running the analysis four times, once for each of the four starting approaches previously evaluated, and selecting the best run of the four analyses, the probability of finding the most optimal solution is very high. The performance of the MVP heuristic is therefore judged to be quite satisfactory.

CHAPTER SEVEN

7.0 The TDVP Heuristic

7.1 Introduction

It was noted in the introductory chapter that most travelling salesman problems researched assume reward or demand potential at the nodes to be uniform through time. This implies that the specific time of arrival at any one node is not an explicit consideration in the problem formulation, and the emphasis of solution procedures is to minimise the penalty for travelling between the nodes. It was suggested that there exist however a number of travelling salesman type problems where the reward potential at the different nodes has an underlying temporal demand function. In such instances, the assumption of uniform reward through time is no longer realistic, and solution procedures, such as the previously discussed MVP heuristic, can no longer identify the most optimal solution.

Given a problem that contains an underlying temporal demand function, a problem formulation and a solution approach is required that considers the time of arrival at the different nodes explicitly. The formulation of this type of a problem has already been discussed in section 4.3. The purpose of this chapter is to derive a possible solution approach to such problems, that of the TDVP heuristic.

The chapter commences by outlining the logical procedures underlying the TDVP heuristic. Subsequently, a number of time dependent demand functions potentially applicable to travelling salesman type problems are discussed. To evaluate performance, a version of the TDVP heuristic containing one of the temporal demand functions outlined has been added as an option within the MVP program. A number of analyses utilising this option are performed and evaluated.

7.2 The TDVP Heuristic

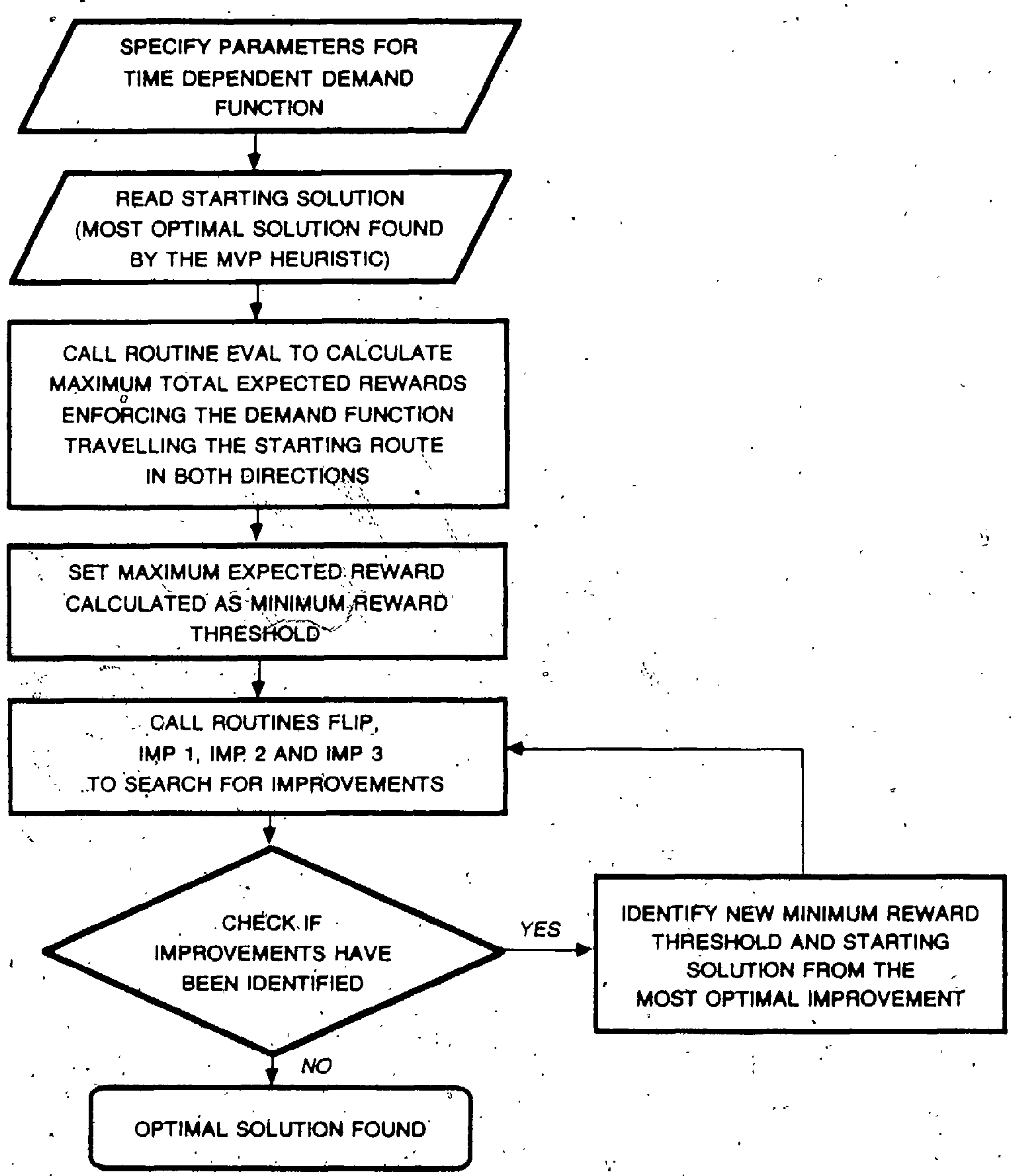
The overall strategy underlying the TDVP heuristic is the same as that underlying the MVP heuristic, of maximising reward within a defined maximum penalty constraint. The TDVP heuristic is therefore based on the same logic and design philosophy as the MVP heuristic. The aim is to maximise reward by continually reshuffling and altering a given route sequence until no further improvement can be found. For a given temporal demand function we suspect intuitively that an optimal solution will have been found when most of the time spent servicing the nodes is clustered at or close to the peak(s) of the time specific demand function, and when most of the necessary time spent travelling the links is when demand is low.

The TDVP heuristic searches for improvements over a given route sequence by utilising four improvement routines. Three of these routines are again based on the principles of node addition, substitution and subtraction. The fourth routine concerns a shuffling process. The overall design of this heuristic is presented as a flowchart in Figure 7.1.

The heuristic takes as a starting solution for a given problem specification the optimal solution derived by the MVP heuristic. Such a solution is, of course, the best possible solution found assuming that reward potentials at the demand nodes are independent of the time of arrival.

The first step in the heuristic procedure is to recalculate the maximum possible reward collected for that starting sequence when

Figure 7.1. The TDVP Heuristic Procedure



enforcing a specified time dependent demand function. This requires the calculation of the time of arrival, the reward potential at that time of arrival, and the length of stay to service that reward for each node within the given route sequence. These calculations are performed by routine EVAL.

Routine EVAL is based on the following logical procedure: The user must first specify a time at which routing is to start, T_{START}, and the time when reward peaks, T_{PEAK}. Routine EVAL treats T_{PEAK} as time 0. It proceeds by calculating all times relative to this peak. The routine thus first calculates T_{START} relative to this peak. This time is identified by A_{0_1}, the time of arrival at 0_1, the depot. Departure from the depot is immediately after arrival, and is therefore also equal to A_{0_1}. The time of arrival at the first node to be visited from the depot, D_1 or 0_2, is defined as A_{0_2}. A_{D_2} equals $A_{0_1} + P_{0_1,D_1}$ where P_{0_1,D_1} is the penalty required to travel the first link in the route. To allow for these calculations, penalty must be measured in units of time.

Once A_{0_2} has been obtained, routine EVAL proceeds to calculate the reward that can be expected at 0_2 at the time of arrival, A_{0_2}. This reward potential is identified as $REX_{0_2,A_{0_2}}$. Once the expected reward potential has been identified, the routine will evaluate the length of time required to collect this reward. It is assumed that the reward potential of a node is dependent only on the time of arrival. The length of time spent at the node is therefore a linear function of reward.

Given this assumption, the time spent servicing a node can be calculated in the same way as for the MVP heuristic, that is by multiplying the expected reward potential by the constant Q (the time required to service one unit of demand).

The time of departure from a node i is the sum of the time of arrival at that node and the time spent servicing that node's reward potential. The time of arrival at the next node, $A_{O_{i+1}}$, is therefore defined as follows:

$$A_{O_{i+1}} = A_{O_i} + (Q * REX_{O_i A_{O_i}}) + P_{O_i O_{i+1}} \tag{7.1}$$

Utilising the steps outlined above, routine EVAL proceeds to calculate the time of arrival, the reward that can be expected at that time and the time of departure from each node in a specified route sequence.

Every time a new node is reached, the routine checks whether the total penalty (time) accumulated so far exceeds PMAX, the specified maximum acceptable penalty threshold. Should it be found that the route sequence presently evaluated exceeds PMAX, then routine EVAL will reject it as infeasible, and will return to the main program of the heuristic. By definition, this situation cannot arise for the starting sequence passed down by the MVP heuristic. This is true for the following reason: The route sequence passed down by the MVP heuristic must have satisfied the PMAX constraint. Total penalty for travelling the links remains the same as calculated by the MVP heuristic when imposing a temporal demand function. Total penalty to be accepted for servicing the nodes calculated by the MVP heuristic assumes that maximum reward is collected everywhere. When imposing a temporal demand function, expected reward and therefore the penalty for collecting it can at most be the same as that for maximum possible total reward. Total penalty can therefore not exceed PMAX for this sequence.

The reason for incorporating the test outlined above will become obvious further on in the discussion. Briefly stated, it is because routine EVAL is called upon to evaluate route sequences many more times further on in the procedure of the TDVP heuristic.

Routine EVAL considers travel to be possible in a clockwise and an anti-clockwise direction for any one given route sequence. It therefore evaluates maximum reward to be expected from a given route sequence for a specific temporal demand function twice, once for the route sequence passed down, once for the inverse of that sequence. When evaluating the starting sequence, routine EVAL will therefore identify which direction of travel for that sequence offers the largest possible reward. The routine will set this largest possible reward potential as a mimimum expected reward threshold. It will store the starting sequence in memory as the temporary best identified solution, adjusting the direction of travel if necessary.

The TDVP heuristic next calls upon a number of improvement routines, whose objective is to identify new route sequences whose total reward potential may exceed the minimum reward threshold defined above. Each new route sequence identified is therefore evaluated by routine EVAL as outlined before. Every time routine EVAL detects a route sequence that exceeds the minimum reward threshold, subject to remaining within the limits imposed by PMAX, it is substituted and stored as the new best route sequence found. The minimum reward threshold is adjusted to equal the total reward potential for this new route.

The first improvement routine called by the TDVP heuristic is routine FLIP, a shuffling routine. Routine FLIP commences by taking every node incorporated in the present route in turn, (with the exception of the depot), and attempts to insert it somewhere else within the route. This procedure is conceptually similar to routine SWITCH in the MVP heuristic. As already mentioned, each new route sequence derived is evaluated by routine EVAL.

Routine Flip next takes every adjacent pair of nodes (not including the depot), and tries to place this adjacent pair somewhere else within the route. Every time an adjacent pair of nodes is placed somewhere else, it is inserted twice, first by maintaining the direction of travel between the two nodes, second by reversing the direction of travel. As an example, if adjacent nodes **a** and **b** were to be inserted between nodes **c** and **d**, the route segments **c-a-b-d** and **c-b-a-d** would result.

Routine FLIP next takes every set of three adjacent nodes in turn (not including the depot), and proceeds to place these sets elsewhere within the route as outlined previously. The routine continues to try and improve on an existing sequence by attempting to shuffle increasingly larger strings of adjacent nodes within a route. The largest set considered are strings that contain n/2 members.

After one complete run of routine FLIP, the heuristic will check whether any improvements were detected. If yes, then the new best solution, presently stored by routine EVAL, is entered as a new solution to routine FLIP, and the entire shuffle is re-run. This process is repeated until a route sequence has been identified for which routine

FLIP cannot detect an improvement. In this case, the heuristic moves on and calls routine IMP1.

Routine IMP1 is based on a ONE IN - ZERO OUT addition. It is conceptually similar to routine IMPROV1 incorporated within the MVP heuristic, and will therefore not be discussed in any detail here. It should suffice to know that every time routine IMP1 identifies a new route sequence, it again is evaluated by routine EVAL. Routine IMP1 is re-run until it can detect no further improvements over a given route. Once this occurs, the heuristic moves on and calls routine IMP2.

Routine IMP2 is based on node substitution, a ONE IN - ONE OUT routine. It is conceptually similar to routine IMPROV2 within the MVP heuristic. This routine, too, relies on routine EVAL to evaluate every new route sequence identified, and is re-run until it cannot detect any further improvement over a given route.

The last improvement routine called is routine IMP3, a ONE IN - TWO OUT routine. It is conceptually similar to routine IMPROV3 within the MVP heuristic. As was noted for IMPROV3, IMP3 only considers adjacent pairs of nodes to be substituted out.

Once routine IMP3 cannot identify any further improvements, the TDVP heuristic proceeds to check whether any one of the four routines identified an improvement for this run. If so, then the heuristic returns to call routine FLIP, and the entire search is repeated for the new best route. If none of the four routines identified an improvement, then the most optimal solution capable of being detected by the heuristic has been found. In this case the heuristic will proceed by printing the optimal route sequence. It will further print total reward expected, total penalty accrued, and the time of arrival, expected reward potential and length of stay for each node.

7.3 Time Dependent Demand Functions

Before proceeding to incorporate the TDVP heuristic within the MVP program in order to derive and evaluate results for a number of analyses, it is first necessary to elaborate more specifically on some time dependent demand functions.

Three time dependent demand functions are discussed below. They are a quadratic, an oscillating and an approximation of a gaussian type function respectively. Examples of the shapes underlying these three class of functions are presented in Figure 7.2.

7.3.1 A Quadratic Time Function

The general equation for a quadratic function is the following:

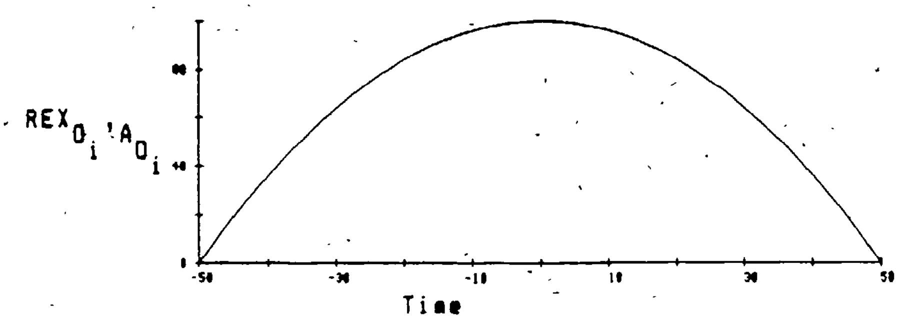
$$Y = a + b*X + c*X^2 \tag{7.2}$$

The general shape is that of a parabola. It contains one turning point identified by differentiation, and for infinitely large positive or negative values of X, values of Y are at negative or positive infinity, depending on whether the turning point represents a maximum or a minimum respectively.

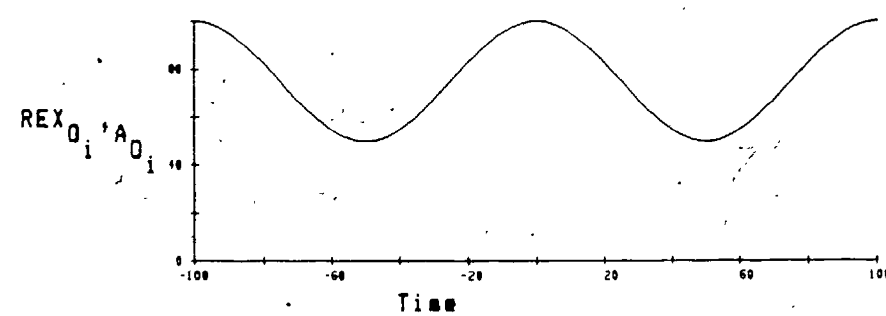
A quadratic temporal demand function can be approximated by the following equation:

Figure 7.2 Examples of Time Dependent Demand Functions

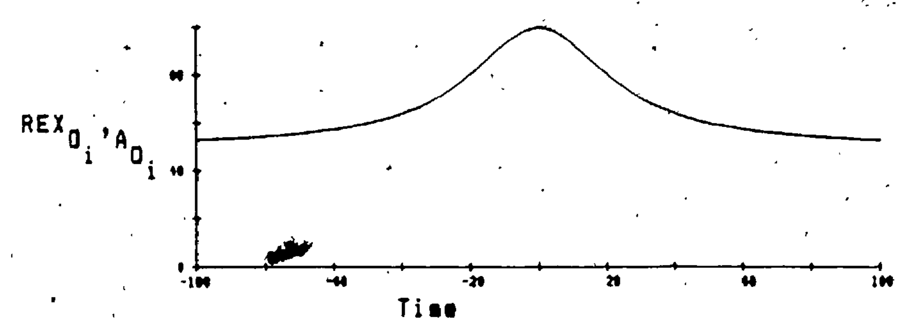
a) Shape of a quadratic time function:
Parameters: $R_{0_i} = 100$, $T_{MIN} = 50$



b) Shape of an oscillating time function
Parameters: $R_{0_i} = 100$, $c = 0.75$, $T_{CYCLE} = 50$



c) Shape of an approximation of a gaussian type time function
Parameters: $R_{0_i} = 100$, $c = 0.5$, $T_{HALF} = 25$



$$REX_{O_i,A_{O_i}} = R_{O_i} * \left(1 - \frac{A_{O_i}^2}{T_{MIN}^2}\right) \qquad 7.3$$

$REX_{O_i,A_{O_i}}$ stands for the reward that can be expected at node O_i at the time of arrival, A_{O_i}. As already noted in section 7.2, A_{O_i} is measured relative to time 0 at which reward peaks. It can be deduced from equation 7.3 that at $A_{O_i} = 0$ the value for $REX_{O_i,A_{O_i}} = R_{O_i}$, the maximum possible reward for node i. T_{MIN} stands for the time prior to or after t_0 when the reward potential turns zero. The collection of reward is thus only feasible when $-T_{MIN} < A_{O_i} < T_{MIN}$. An example of a quadratic time dependent demand function is shown in Figure 7.2.a. To depict the shape of this function, it is assumed that a vendor arrives at a node O_i continually between $-T_{MIN}$ and T_{MIN}. The shape shown in Figure 7.2a was derived when setting $R_{O_i} = 100$ for $T_{MIN} = 50$.

An example of a quadratic temporal demand function in the vending context is that of vending to service the demand of people attending sport events. Early attendees will arrive at some time prior to the start of the event, representing potential demand for the vendor's services. Demand potential will peak during the actual event, and it will return to zero when all attendees have left. A more general example of a routing problem with one peak, and that is confined to a limited time period, is that of a candidate canvassing for votes, an example already discussed in the introductory chapter. Here, the candidate's route is confined to the official campaign period, and the function will peak just previous to the election.

7.3.2 An Oscillating Time Function

An example of an oscillating time function is that of a cosine function as shown below:

$$Y = a + \cos \frac{X}{b} \qquad 7.4$$

This type of a function will oscillate within a specified amplitude, reaching a peak at specified intervals equal to the length of one complete cycle of a cosine wave. The oscillating time dependent demand function shown in Figure 7.2b was derived from the following equation:

$$REX_{O_i A_{O_i}} = c * R_{O_i} + \left[\left(R_{O_i} - \left(c * R_{O_i} \right) \right) * \left(\cos \frac{A_{O_i}}{T_{CYCLE}} \right) \right] \qquad 7.5$$

Here, c stands for the average expected percentage of total demand potential at node O_i (see Figure 7.2b). Thus for c = .75, the average demand to be expected equals 75 percent of R_{O_i}, the maximum possible demand. T_{CYCLE} stands for the length ot time required to complete one cycle of the cosine curve. Reward will therefore peak at $A_{O_i} = 0$ and at $A_{O_i} = k*T_{CYCLE}$, where k stands for any integer multiple of 2π. Figure 7.2b was derived by continually arriving at a node O_i between $A_{O_i} = -100$ and $A_{O_i} = 100$. R_{O_i} was specified as 100, c as 0.75 and T_{CYCLE} as 50.

An example of an oscillating demand function in the vending context is that of the seasonal demand for a product, such as the change in demand for ice cream throughout the year. More generally, an oscillating demand function applies to the periodic marketing of consumer products such as bread or soap. Here, demand will replenish itself at set intervals.

7.3.3 An Approximation of a Gaussian Time Function

The mathematical definition for a Gaussian function is the following:

$$Y = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(x-\mu\right)^2 / \sigma^2} \qquad 7.6$$

μ stands for the mean of the given distribution, σ for the standard deviation. The shape of this function is that of a bell. It contains one peak and tails off to reach Y values of zero for infinitely large negative and positive values of X. An approximation of a Gaussian distribution can be obtained from the following equation:

$$REX_{O_i A_{O_i}} = c * R_{O_i} + \left(\frac{1 - c}{1 + (A^2_{O_i} / T^2_{Half})} \right) * R_{O_i} \qquad 7.7$$

Here, c stands for the percentage of maximum reward that can always be expected (see Figure 7.2c). For infinitely large negative or positive values of A_{0_i}, $Rex_{0_i A_{0_i}}$ will therefore equal 50 percent if c = 0.5. T_{HALF} stands for the time prior to and after the peak at which the reward potential at a node is expected to be halfway between the maximum and the minimum expected reward.

Figure 7.2c shows a time dedendent demand function based on equation 7.7 when pretending to arrive at a node i continually between $A_{0_i} = -100$ and $A_{0_i} = 100$ for c = 0.5, a maximum reward of 100 and T_{HALF} at 25.

An example of this type of temporal demand function in the vending context is that of a salesman selling souvenirs of the Pope to stores. There will always be some demand for articles displaying the image of the Pope, but demand will surge around the time of a Papal visit. More generally, this type of demand function applies to the vending of any product for which there exists a general demand, but whose demand increases throughout the occurrence of a special event.

7.4 Testing the TDVP Heuristic

To evaluate the TDVP heuristic, it has been incorporated within the MVP program. For the purpose of its evaluation the temporal demand function utilised is that of an approximation of a Gaussian type function as discussed above.

It is possible to incorporate the TDVP heuristic as an option within the MVP program so that it is called every time an optimal solution has been identified by any one of the three procedure options outlined in section 5.1. This would allow the derivation of noninferior solution sets and surfaces for multiobjective travelling salesman type routing problems with an underlying temporal demand function. The TDVP heuristic is however computationally very time consuming. Incorporating the heuristic within Procedures Two or Three would therefore imply potentially very substantial increases in the cost of running the MVP program.

To avoid such a substantial increase in cost when evaluating the performance of the heuristic, it has been incorporated within the MVP program after Procedure One. As already discussed in section 5.1, Procedure One calls the MVP heuristic to identify one optimal solution for a specified set of parameters, given one maximum acceptable penalty threshold. This optimal solution can function as the starting solution to the TDVP heuristic. Alternatively, it is possible to alter this starting solution to evaluate the heuristic's performance for different initial input of route sequences by utilising option 3 of the MVP program (see section 5.1).

To perform TDVP analyses, the user must first specify parameters that define the time specific demand function. For an approximation of a Gaussian function these parameters are the following:

c - the minimum percentage of maximum total reward that can always be expected when arriving at a node, where $0 \leq c < 1$

T_{START} - the time when routing is to start

T_{PEAK} - the time at which expected reward potential equals the maximum reward potential

T_{HALF} - the time at which expected reward potential is halfway between the minimum amd the maximum expected reward

To test the heuristic the West German problem discussed in Chapter Six was again utilised. Two scenarios are evaluated in some detail.

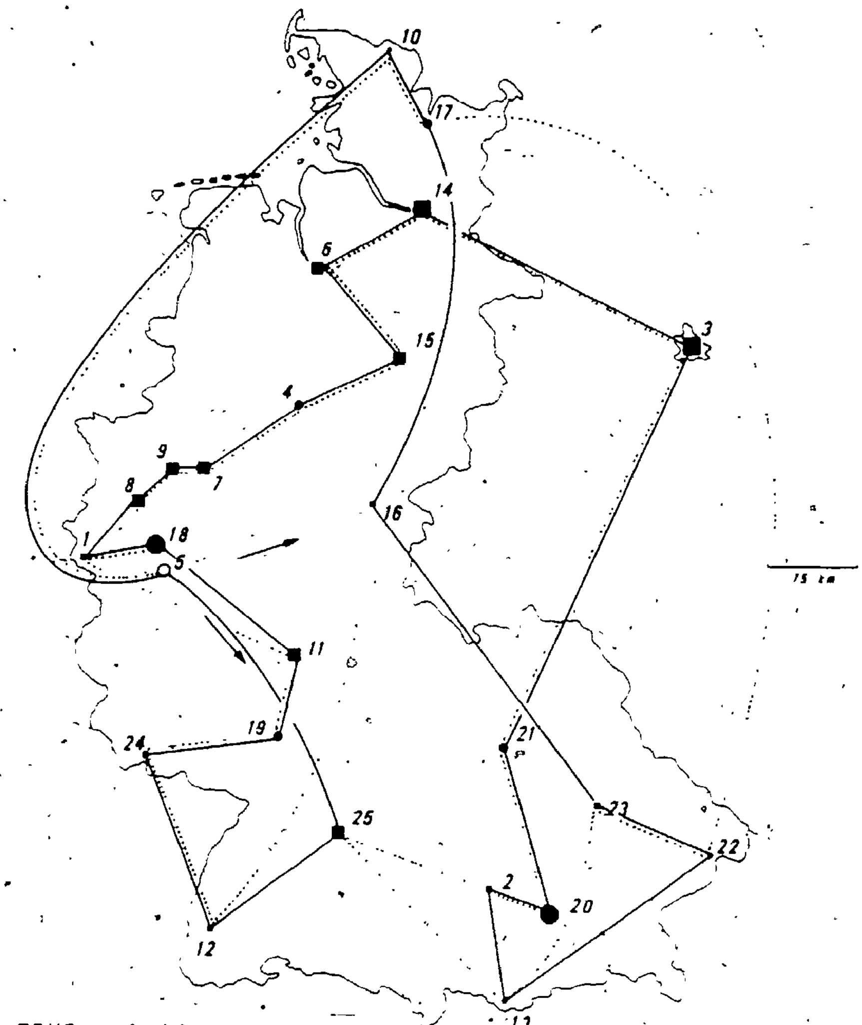
The first scenario concerns finding the optimal route that will connect all 25 cities for a specified temporal demand function. This function was defined by setting $T_{PEAK} = 7000$, $T_{HALF} = 3500$ and $T_{START} = 0$. PMAX was defined as very large and Q = 1.0. To evaluate how the solution for these specifications would change for different values of c (the minimum percentage of maximum reward always expected), the analysis was performed three times setting c to 0.8, 0.5 and 0.0. For each value of c, the analysis was re-run 10 times. The first run utilised as a starting solution the optimal solution found by the MVP heuristic. For the subsequent nine runs, the MVP derived starting sequence was modified utilising option 3 of the MVP program. Five of these nine runs were deliberately modified so that time spent servicing the nodes occurs at low demand, while time spent routing the links occurs at high demand. These five runs therefore deliberately commence from very poor starting sequences. The remaining four starting sequences were defined to be intuitively better than the optimal route sequence so far discovered.

Tables 7.1 to 7.3 present detailed information about the best and worst solutions obtained. Figures 7.3 to 7.5 show the respective optimal routes. Eighteen of the thirty analyses yielded the respective optimal solutions. The MVP defined starting sequence only led to the identification of the best solution once. All of the starting sequences defined that were thought intuitively better than the optimal solution so far discovered proved to be inferior. Most did however lead back to the optimal solution. The percentage differences in total reward between the best and worst solutions found are very small, 0.01 percent, 0.04 percent and 0.32 percent for c equals 0.8, 0.5 and 0.0 respectively. As expected, all three best routes cluster nodes with high maximum reward

Table 7.1 Route Sequences Derived for $c = 0.8$, $P_{Max} = \infty$, $T_{Peak} = 7000$, $T_{Half} = 3500$ and $T_{Start} = 0$

Most optimal MVP solution found	Most optimal TDVP solution derived				Worst TDVP solution derived			
Route	Route	A_{S_i}	% R_{S_i}	REX_{S_i}, A_{S_i}	Route	A_{S_i}	% R_{S_i}	REX_{S_i}, A_{S_i}
5	5	-7000.0	-	-	5	-7000.0	-	-
18	25	-6653.0	84.3	501.8	16	-6735.0	84.3	171.9
1	12	-5974.2	85.1	148.2	12	-6090.1	85.0	147.8
8	24	-5591.1	85.6	175.5	24	-5734.3	85.4	175.1
9	19	-5287.5	86.1	268.6	19	-5431.2	85.9	267.9
7	11	-4936.9	86.7	547.0	11	-5681.2	86.4	545.4
4	18	-4199.9	88.2	891.7	1	-4276.8	88.0	213.0
16	1	-3239.2	90.8	219.7	18	-3994.8	88.7	896.6
15	8	-2944.5	91.7	604.4	8	-3050.2	91.3	602.1
6	9	-2307.2	93.9	633.2	9	-2415.1	93.5	630.5
10	7	-1638.0	96.4	605.4	7	-1748.6	96.0	602.9
17	4	-920.6	98.7	310.9	4	-1033.7	98.4	309.9
14	15	-495.6	99.6	547.8	15	-609.7	99.4	546.8
3	6	165.2	99.9	570.7	6	50.0	99.9	571.0
21	14	854.9	98.9	1688.8	14	740.0	99.1	1693.4
23	3	2833.7	92.1	1811.2	3	2723.4	92.5	1818.6
22	21	5085.9	86.4	428.7	21	4983.0	96.6	429.6
20	20	5679.6	85.5	1121.8	20	5577.6	85.7	1123.7
13	2	6867.4	84.1	208.6	2	6767.3	84.2	208.9
2	13	7195.1	83.8	25.1	25	7137.2	83.9	499.1
25	22	7507.2	83.6	33.4	13	7916.3	83.3	25.0
12	23	7661.6	83.5	109.3	23	8161.3	83.1	108.9
24	16	8200.9	83.1	169.5	22	8391.1	82.7	33.2
19	17	8786.5	82.7	215.9	17	9381.3	82.5	215.2
11	10	9085.4	82.6	41.3	10	9679.5	82.3	41.2
5	5	9771.7	82.3	233.7	5	10365.6	82.0	233.0
Total Reward				12112.4				12110.7
Total Penalty				17005.4				17598.7

Figure 7.3 The Route Sequences Shown in Table 7.1



——— = Optimal TDVP solution:
5 25 12 24 19 11 18 1 8 9 7 4 15 6 14 3 21 20 2 13 22 23 16 17 10 5

·········· = Worst derived TDVP solution:
5 16 12 24 19 11 1 18 8 9 7 4 15 6 14 3 21 20 2 25 13 23 22 17 10 5

For identification of nodes refer to Table 6.1.

○ = Depot

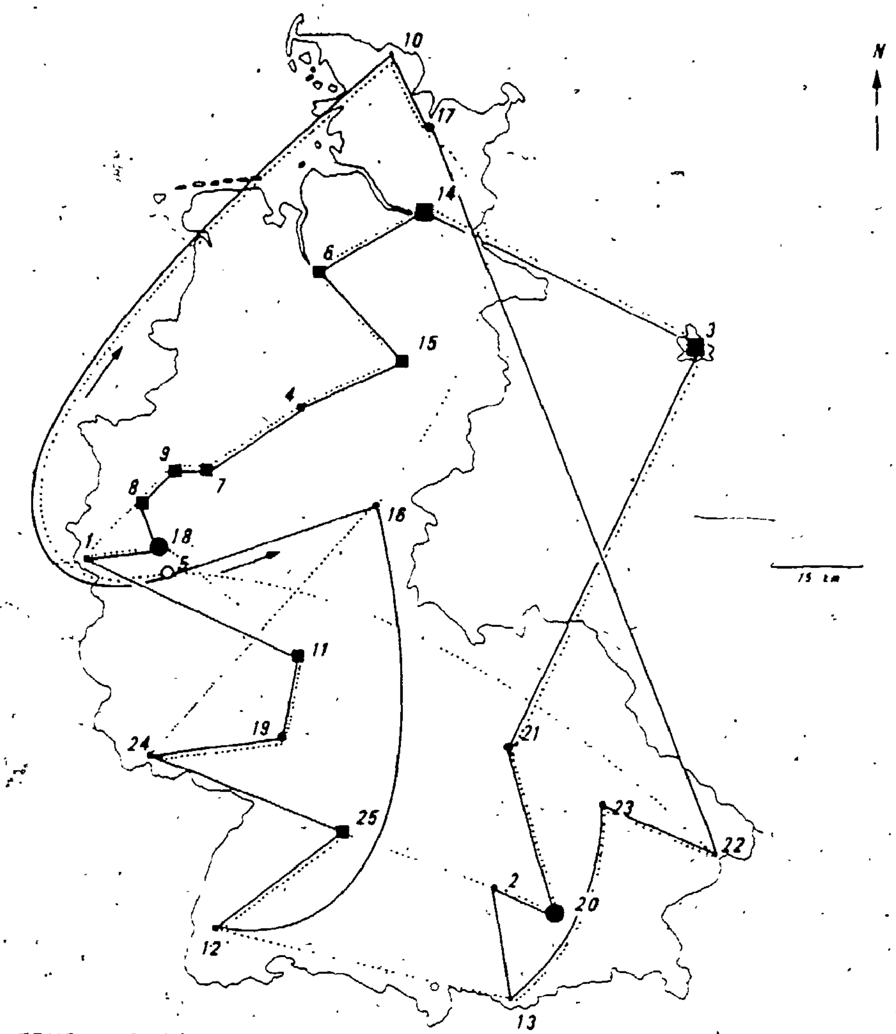
Size of Population (/1000)

· = 0 - 99	• = 100 - 249	● = 250 - 499
■ = 500 - 999	⬤ = 1000 - 1500	◼ = > 1500

Table 7.2 Route Sequences Derived for $c = 0.5$, $P_{Max} = \infty$, $T_{Peak} = 7000$, $T_{Half} = 3500$ and $T_{Start} = 0$

Most optimal MVP solution found	Most optimal TDVP solution derived				Worst TDVP solution derived			
Route	Route	A_{S_i}	% R_{S_i}	REX_{S_i},A_{S_i}	Route	A_{S_i}	% R_{S_i}	REX_{S_i},A_{S_i}
5	5	-7000.0	-	-	5	-7000.0	-	-
18	16	-6735.0	60.6	123.7	10	-6354.0	61.6	30.8
1	12	-6138.3	62.3	108.3	17	-6240.2	62.0	161.7
8	25	-5825.9	63.4	376.4	16	-5662.5	63.8	130.2
9	24	-5210.6	65.6	134.4	24	-5138.3	65.9	135.0
7	19	-4948.2	66.7	208.0	19	-4875.3	67.0	209.1
4	11	-4658.2	68.0	429.3	11	-4584.2	68.4	431.7
16	1	-3969.8	71.9	173.9	18	-3962.5	71.9	727.0
15	18	-3726.9	73.4	742.4	1	-3166.5	77.5	187.5
6	8	-2936.5	79.3	522.9	8	-2903.9	79.6	524.7
10	9	-2380.6	84.2	567.4	9	-2346.3	84.5	569.5
17	7	-1777.2	89.2	563.6	7	-1740.8	90.1	565.7
14	4	-1101.6	95.9	300.8	4	-1063.1	95.8	301.7
3	15	-686.8	98.1	539.8	15	-674.4	98.4	540.9
21	6	-34.0	99.9	571.0	6	6.5	99.9	571.0
23	14	656.0	98.3	1679.0	14	696.5	98.1	1675.5
22	3	2624.9	82.0	1612.9	3	2662.0	81.7	1606.6
20	21	4678.9	67.9	337.0	21	4709.6	67.8	336.2
13	20	5180.9	65.7	861.6	20	5210.8	65.5	859.9
2	2	6108.5	62.4	154.6	2	6136.8	62.3	154.4
25	13	6382.1	61.6	18.5	25	6452.2	61.4	365.1
12	23	6620.6	60.9	79.8	12	7021.3	60.0	104.3
24	22	6821.4	60.4	24.2	13	7473.7	58.9	17.7
19	17	7802.6	58.4	152.4	23	7711.3	58.5	76.7
11	10	8037.9	58.0	29.0	22	7909.0	58.2	23.3
5	5	8711.9	56.9	161.7	5	8537.3	57.2	162.4
Total Reward				10472.7				10468.8
Total Penalty				15873.7				15698.8

Figure 7.4 The Route Sequences Shown in Table 7.2



——— = Optimal TDVP solution:
5 16 12 25 24 19 11 1 18 8 9 7 4 15 6 14 3 21 20 2 13 23 22 17 10 5
.... = Worst derived TDVP solution:
5 10 17 16 24 19 11 18 1 8 9 7 4 15 6 14 3 21 20 2 25 12 13 23 22 5

For identification of nodes refer to Table 6.1.

o = Depot

Size of Population (/1000)

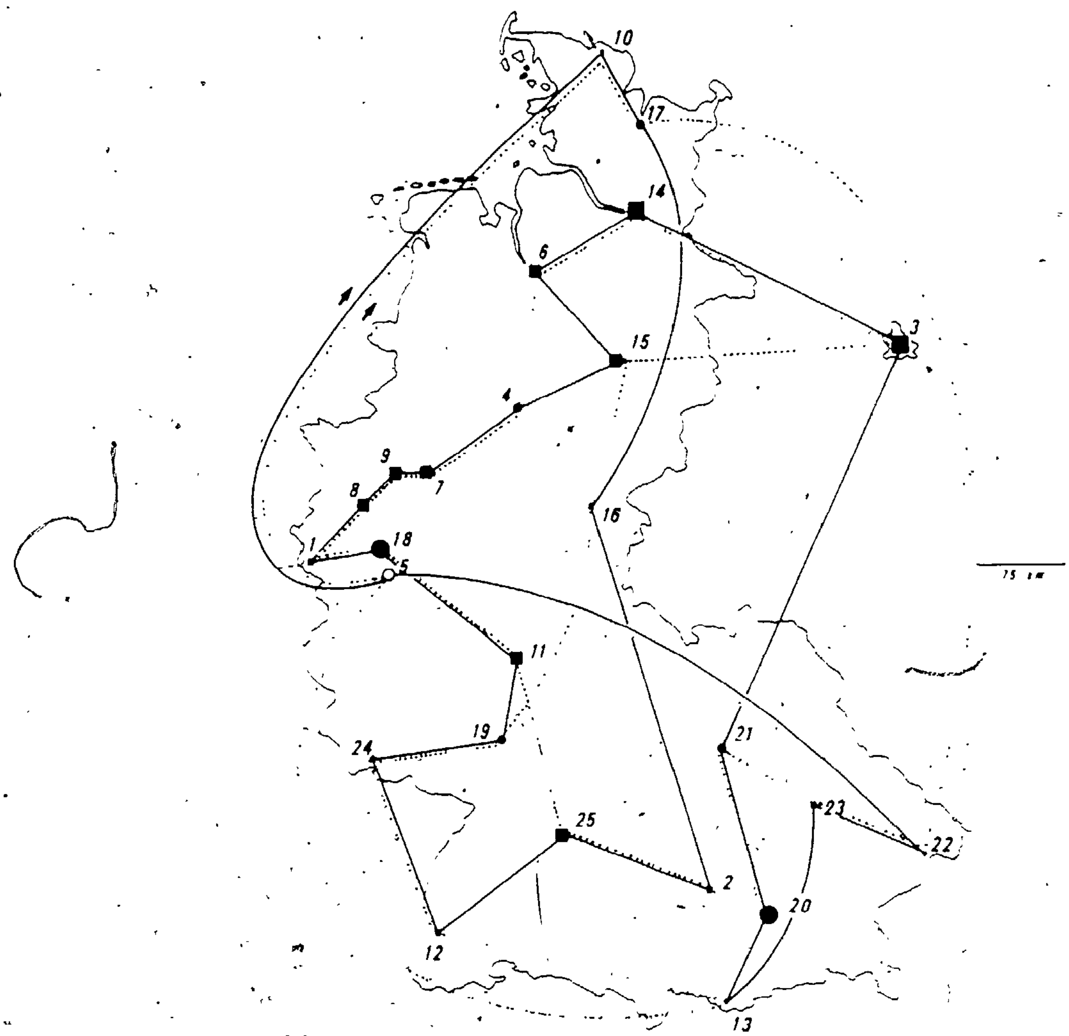
• =	0 - 99	• =	100 - 249	• =	250 - 499
■ =	500 - 999	● =	1000 - 1500	■ =	> 1500

153

Table 7.3 Route Sequences Derived for $c = 0.0$, $P_{Max} = \infty$, $T_{Peak} = 7000$, $T_{Half} = 3500$ and $T_{Start} = 0$

Most optimal MVP solution found	Most optimal TDVP solution derived				Worst TDVP solution derived			
Route	Route	A_{S_i}	% R_{S_i}	REX_{S_i},A_{S_i}	Route	A_{S_i}	% R_{S_i}	REX_{S_i},A_{S_i}
5	5	-7000.0	-	-	5	-7000.0	-	-
18	10	-6355.0	23.3	11.6	10	-6355.0	23.3	11.6
1	17	-6260.4	23.8	62.2	17	-6260.4	23.8	62.2
8	16	-5782.2	26.8	54.7	22	-5241.2	30.9	12.3
9	2	-5293.5	30.4	75.4	23	-5107.9	32.0	41.9
7	25	-5057.1	32.4	192.7	21	-4962.0	33.2	164.8
4	12	-4660.4	36.1	62.8	20	-4632.2	35.7	467.8
16	24	-4389.6	38.9	79.7	2	-4089.4	42.3	104.9
15	19	-4181.9	41.2	128.5	25	-3823.6	45.6	271.3
6	11	-3971.4	43.7	275.8	11	-3357.3	52.1	328.6
10	18	-3505.6	49.9	504.7	18	-2838.7	60.3	609.8
17	1	-2931.9	58.8	142.2	1	-2159.8	72.4	175.3
14	8	-2714.7	62.4	411.5	8	-1909.6	77.1	507.8
3	9	-2270.2	70.4	474.4	9	-1368.7	86.7	584.6
21	7	-1759.8	79.8	501.3	7	-748.2	95.6	600.6
23	4	-1146.5	90.3	284.5	4	-35.6	99.9	314.9
22	15	-748.1	95.6	526.0	6	432.4	98.5	562.4
20	6	-109.1	99.9	570.4	14	1113.8	90.8	1550.9
13	14	580.0	97.3	1662.3	3	2954.7	58.4	1148.5
2	3	2532.0	65.6	1291.0	15	4388.2	38.9	213.9
25	21	4264.7	40.2	199.6	16	4767.1	35.0	71.5
12	20	4629.3	36.4	477.2	19	5117.5	31.9	99.4
24	13	5204.5	31.1	9.3	24	5344.9	30.0	61.5
19	23	5433.8	29.3	38.4	12	5614.5	28.0	48.7
11	22	5593.2	28.1	11.3	13	6011.2	25.3	7.6
5	5	6209.5	24.1	68.5	5	6649.8	21.6	61.6
Total Reward				7378.2				7275.4
Total Penalty				13278.0				13711.4

Figure 7.5 The Route Sequences Shown in Table 7.3



——— = Optimal TDVP solution:
5 10 17 16 2 25 12 24 19 11 18 1 8 9 7 4 15 6 14 3 21 20 13 23 22 5

------- = Worst derived TDVP solution:
5 10 17 22 23 21 20 2 25 11 18 1 8 9 7 4 6 14 3 15 16 19 24 12 13 5

For identification of nodes refer to Table 6.1.

o = Depot

Size of Population (/1000)

· = 0 - 99 • = 100 - 249 • = 250 - 499
■ = 500 - 999 ● = 1000 - 1500 ■ = > 1500

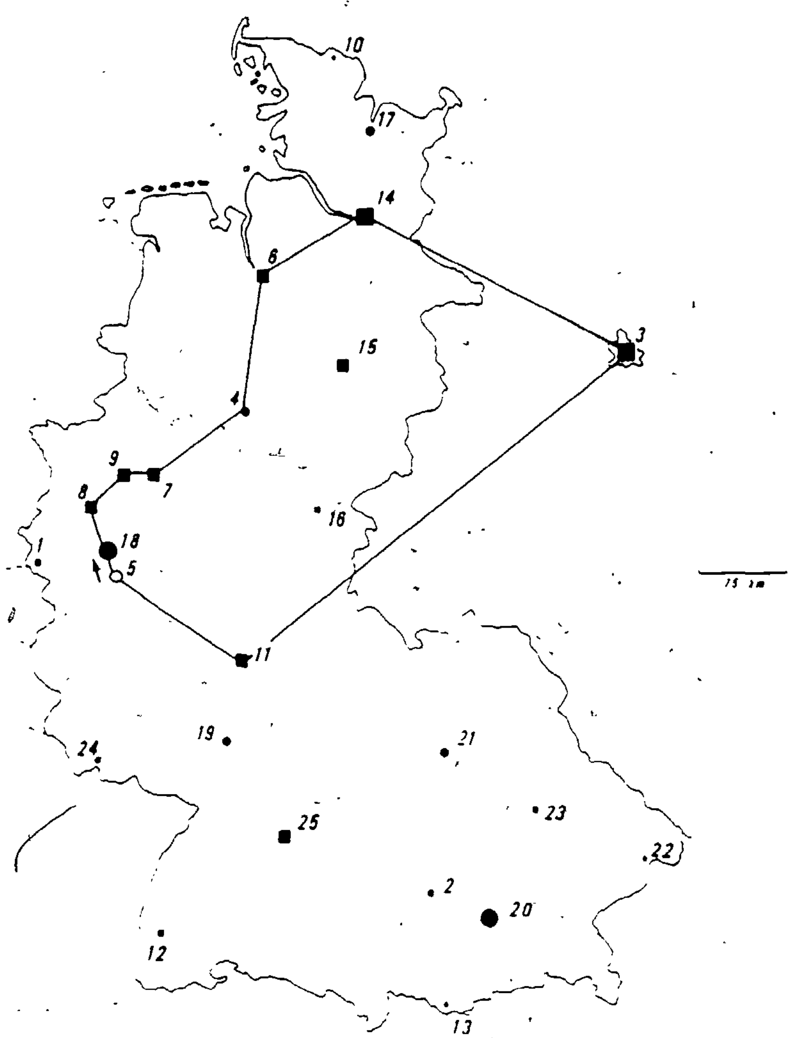
potential into the peak, and nodes with low potential into the tails of the temporal demand function.

The second scenario evaluated concerns that of imposing a maximum time constraint, that of PMAX = 10,000. The optimal solution to this problem found by the MVP heuristic for Q = 1.0 is shown in Figure 7.6. This problem was re-evaluated imposing three different Gaussian temporal demand functions. All three functions start routing at T_{START} = 0 and c = 0.5. The difference between T_{PEAK} and T_{HALF} in all three cases equals 1000. The difference between the three functions therefore lies in the definition of T_{PEAK}. T_{PEAK} is set to 2500, 5000 and 7500 implying that reward peaks soon after routing starts, halfway through the route, and towards the end of the route respectively.

Each function was evaluated ten times as described for the evaluation of the first scenario. The optimal solution was found eight times out of thirty. The MVP defined starting solution again identified the optimal solution once. This time, two of the twelve starting sequences thought intuitively better than the best solution so far identified led to the discovery of superior solutions. Table 7.4 yields detailed information concerning the three best solutions identified. The associated routes are shown in Figures 7.7 to 7.9. Again, all three best routes cluster nodes of high reward potential into the peaks, and nodes with low reward potential into the tails of the demand functions.

The optimal route identified for T_{PEAK} = 7500, shown in Figure 7.9, contains what appears to be a quite irrational route segment. This segment concerns travelling from node 11 (R_{MAX}= 631) to node 2 (248) to node 25 (595) and on to node 20 (1312). An intuitively more obvious route segment here would be to route from node 11 to node 2, to node 25

Figure 7.6 The Optimal Solution Found for Q = 0.1 and P_{MAX} = 10,000



Route Sequence: 5 18 8 9 7 4 6 14 3 11 5
Total Reward = 8448.0, Total Penalty = 9997.0

For identification of nodes refer to Table 6.1.

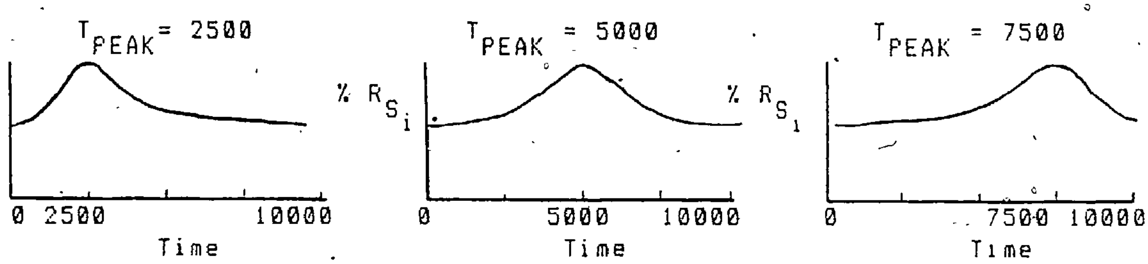
o = Depot

Size of Population (/1000)

·	= 0 - 99	•	= 100 - 249	•	= 250 - 499
■	= 500 - 999	●	= 1000 - 1500	■	= > 1500

Table 7.4 Route Sequences Derived for Different Values of T_{PEAK} When P_{MAX} = 10,000, T_{START} = 0 and c = 0.5

Shapes of the Time Functions:



Characteristics of the Optimal Routes Derived:

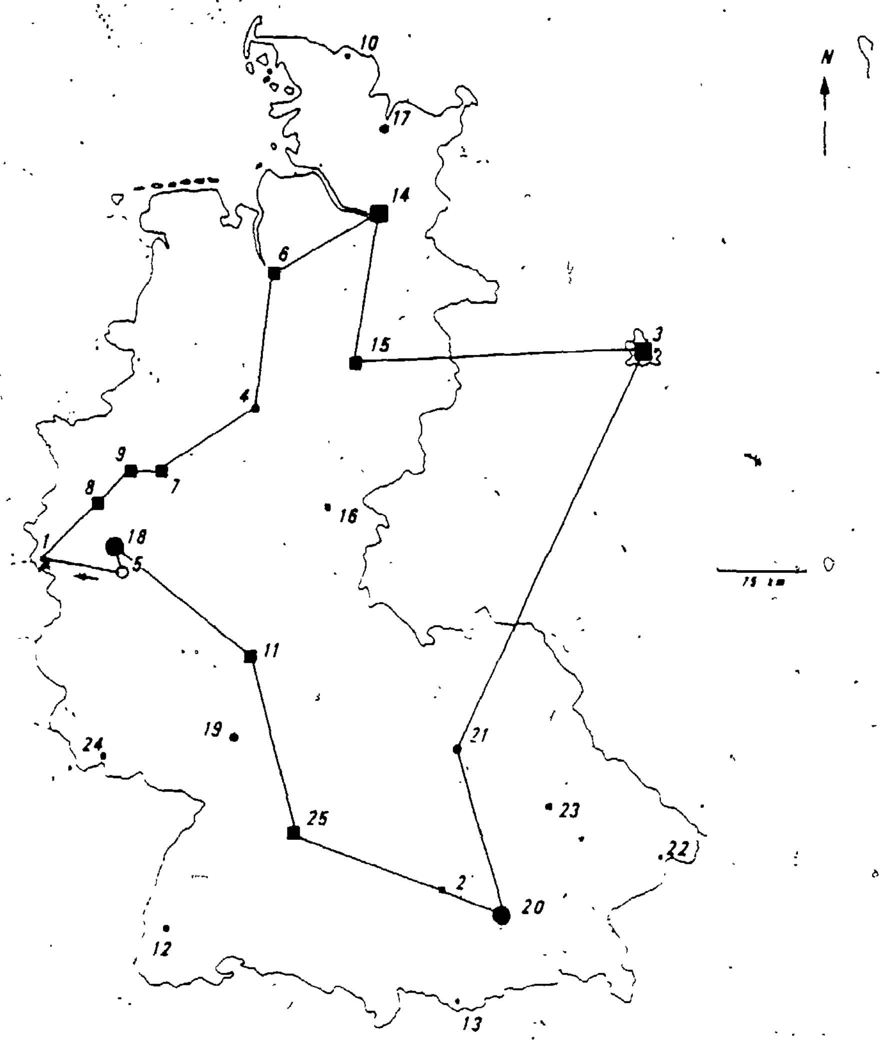
I	II	III	IV	I	II	III	IV	I	II	III	IV
5	0.0	-	-	5	0.0	-	-	5	0.0	-	-
1	90.0	57.3	138.8	2	512.0	52.4	129.9	11	172.0	50.9	321.3
8	303.8	58.6	386.1	20	707.9	52.6	689.8	2	853.3	51.1	126.7
9	722.9	62.0	418.0	21	1562.7	53.9	267.4	25	1141.0	51.2	304.7
7	1176.9	68.2	428.2	18	2228.0	55.7	563.7	20	1665.7	51.4	674.7
4	1717.1	81.0	255.1	1	2861.0	59.0	142.7	21	2505.4	51.9	257.6
6	2125.2	93.8	535.8	8	3078.4	60.7	399.7	3	3204.0	52.6	1034.0
14	2780.0	96.4	1645.9	9	3512.1	65.5	441.8	14	4528.1	55.1	940.9
15	4581.9	59.4	326.6	7	3988.9	74.7	469.3	6	5587.9	60.7	346.8
3	5193.5	56.1	1102.6	6	4709.2	96.1	548.7	15	6047.7	66.1	363.4
21	6737.1	55.7	261.1	14	5376.9	93.8	1601.8	4	6525.1	75.6	238.2
20	7163.2	52.2	684.8	3	7268.7	58.1	1143.5	7	6875.4	86.0	539.9
2	7914.0	51.6	128.1	15	8697.7	53.4	293.7	9	7451.3	99.9	673.2
25	8203.1	51.4	306.4	11	9341.9	52.5	331.4	18	8198.5	83.6	845.2
11	8704.5	51.3	323.5	5	9845.3	52.0	147.8	8	9091.7	64.2	422.7
18	9218.0	51.1	516.5					1	9589.5	59.3	143.5
5	9762.5	50.9	144.6					5	9823.0	57.8	164.2
Total Reward			7602.1				7171.1				7397.2
			(7369.2)*				(6925.7)				(6658.8)
Total Penalty			9907.1				9993.1				9987.2
			(9906.2)*				(9971.7)				(9983.3)

* = Results for the most inferior solution derived

I = Route Sequence II = A_{S_i}

III = % R_{S_i} IV = $REX_{S_i} \cdot A_{S_i}$

Figure 7.7 The Optimal TDVP Solution Found for $Q = 0.1$, $P_{MAX} = 10{,}000$, $T_{START} = 0$, $c = 0.5$ and $T_{PEAK} = 2500$



Route Sequence: 5 1 8 9 7 4 6 14 15 3 21 20 2 25 11 18 5
Total Reward = 7602.1, Total Penalty = 9907.1

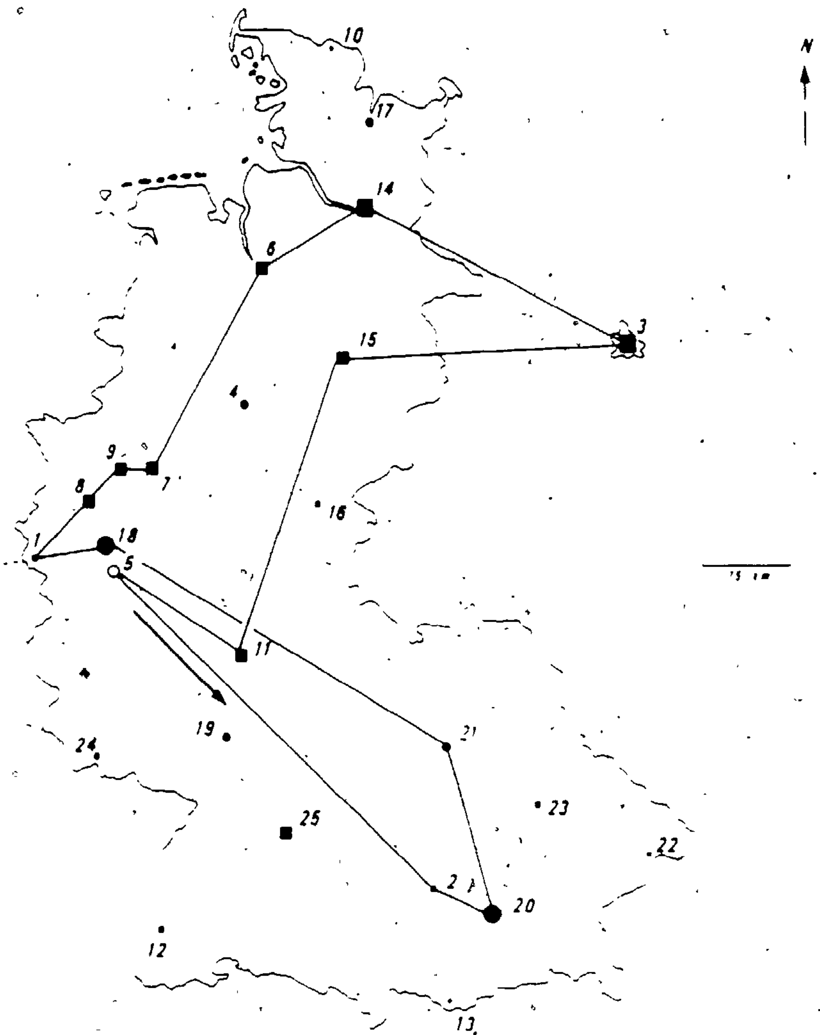
For identification of nodes refer to Table 6.1.

o = Depot

Size of Population (/1000)

·	= 0 - 99	•	= 100 - 249	•	= 250 - 499
■	= 500 - 999	●	= 1000 - 1500	■	= > 1500

Figure 7.8 The Optimal TDVP Solution Found for $Q = 0.1$, $R_{MAX} = 10{,}000$, $T_{START} = 0$, $c = 0.5$ and $T_{PEAK} = 5000$



Route Sequence: 5 2 20 21 18 1 8 9 7 6 14 3 15 11 5
Total Reward = 7171.1, Total Penalty = 9993.1

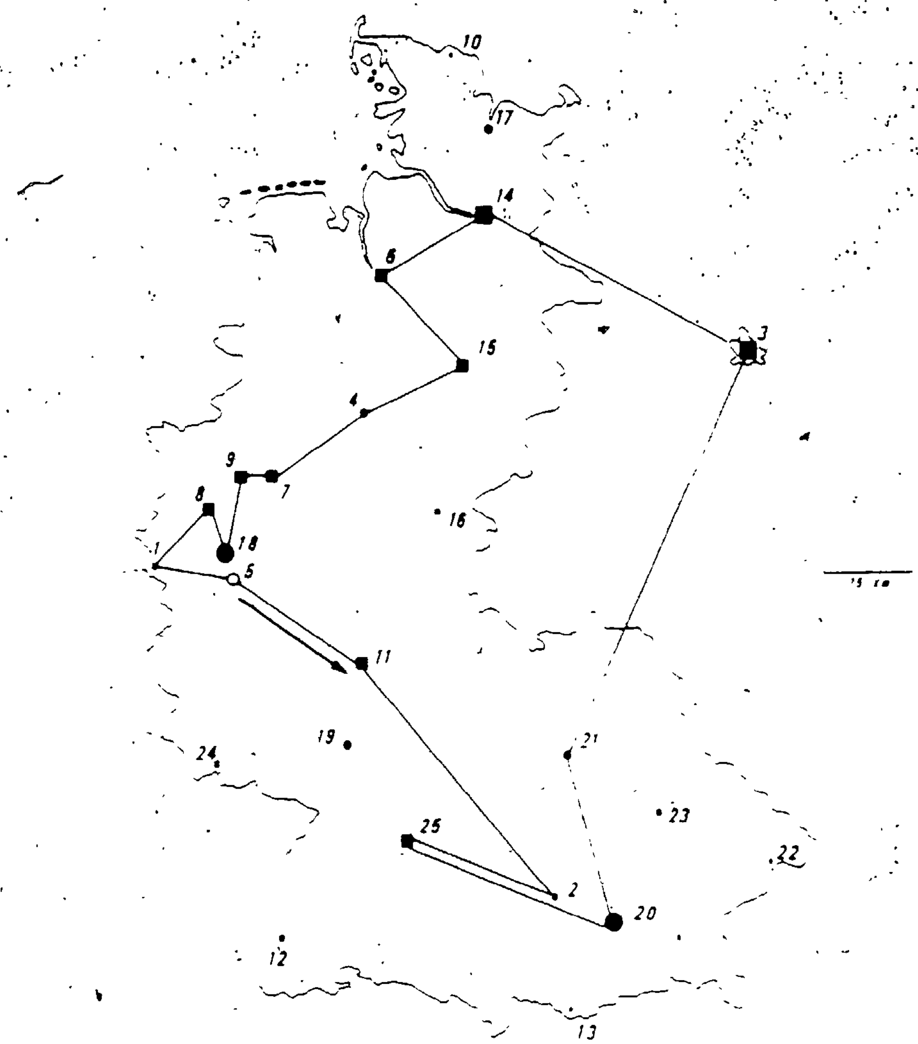
For identification of nodes refer to Table 6.1.

○ = Depot

Size of Population (/1000)

· = 0 - 99	• = 100 - 249	• = 250 - 499
■ = 500 - 999	● = 1000 - 1500	■ = > 1500

Figure 7.9 The Optimal TDVP Solution Found for $Q = 0.1$, $P_{MAX} = 10,000$, $T_{START} = 0$, $c = 0.5$ and $T_{PEAK} = 7500$



Route Sequence: 5 11 2 25 20 21 3 14 6 15 4 7 9 18 8 1 5
Total Reward = 7397.2, Total Penalty = 9987.2

°For identification of nodes refer to Table 6.1.

o = Depot

Size of Population (/1000)

· = 0 - 99	• = 100 - 249	• = 250 - 499
■ = 500 - 999	● = 1000 - 1500	■ = > 1500

and on to node 20. This would imply a 3.4 percent saving in penalty for travelling the links, but would result in a 0.3 percent loss in total reward. Keller (1985) more closely inspects this problem by evaluating the trade off relationship between reward maximisation and penalty minimisation when forced to visit all members of a set of nodes. Given that the overall objective underlying the TDVP heuristic here is to identify a subset of nodes, and the associated route sequence that maximises reward subject to remaining within a maximum acceptable penalty constraint, a 0.3 percent gain in total reward does represent the more optimal solution.

7.5 Discussion

The percentage differences in total reward potential between the best and worst solutions encountered when evaluating the second scenario are 3.06, 3.42 and 9.83 percent for T_{PEAK} = 2500, 5000 and 7500 respectively. These are considerably higher than those encountered when evaluating the first scenario (0.01, 0.04 and 0.28 percent). In addition, the probability of obtaining the optimal solution was found to be 0.60 when evaluating the first scenario, but this probability reduced to 0.24 when solving for the second scenario. These findings lead to the suspicion that the TDVP heuristic performs better when ignoring a maximum penalty constraint, that is when solving for an optimal route sequence for a given temporal demand function that includes all points considered in the analysis. The heuristic appears to perform less well when the entire set cannot be included because of the maximum penalty constraint, that is when the heuristic is forced to select from the overall set a subset of nodes to be included in the route sequence. This suspicion was confirmed when repeating the evaluation for a number of other problems.

An explanation for the above findings becomes obvious when comparing inferior solutions to the best ones found. For the first scenario, inferior solutions tend to differ only marginally from the best in total reward potential. Their route sequences, too, tend to differ little. It can be observed from Tables 7.1 to 7.3 and Figures 7.3 to 7.5 that the respective main differences between the best and the worst routes identified lie in the route segments that occur in the tails of the temporal demand function. The respective route segments at the peak of the functions are very similar in all cases.

These findings are not true for the second scenario. Inferior solutions still tend to yield total reward potentials close to the most optimal found. Inferior route sequences do however tend to be very different from the optimal. This is because the number of combinatorial possibilities of forming a route sequence increase considerably when forced to select a subset of nodes to remain within the maximum penalty constraint. Scenario two type TDVP analyses thus tend to yield solutions only marginally inferior to the best, but with very different subset composition and route sequencing.

Given the occurrence of marginally inferior, yet very different solutions, an extremely complex and time consuming heuristic would be required to identify the optimal route sequence from one which is marginally inferior. The TDVP heuristic already proved to be computationally very time consuming. Given that the added complexities and additional computer time required for a more complex heuristic are not desirable, the TDVP heuristic is thus argued to operate very satisfactorily. It is however recommended that any analysis utilising the TDVP heuristic should be re-run for a number of different starting sequences to increase the probability of identifying the most optimal solution.

CHAPTER EIGHT

8.0 Discussion and Conclusion

8.1 Geography and Optimisation Theory

The advent and growth of 'Quantitative Geography' over the last two decades appears to have become a generally accepted reality within the discipline. Part of this so called 'quantitative revolution' has included geographers expressing more and more of an interest in the application of optimisation theory to asking and solving spatial problems.

Optimisation theory has traditionally, and is continuing to represent an integral part of research undertaken in the discipline of operations research, and in some branches of engineering. The question that arises is whether the geographers' interest in optimisation theory merely represents a duplication, or a borrowing of research efforts undertaken by other disciplines, or whether geographers can claim a unique role in this field of enquiry.

The research emphasis of most engineers and operations researchers involves the search for unique and unambiguous measures of optimality. Mathematical elegance and computational efficiency appear to be the dominant research goals. The applicability of optimisation models to real world problems, and the validity of underlying assumptions represent more minor interests. Without a doubt this theoretical research represents an important and usefull contribution to knowledge. Unique optimal answers do however rarely exist in real world problems. As was already noted in the introductory chapters, real world problems tend to be inherently complex and multiobjective, and one optimal answer will only in exceptional circumstances optimise all the issues.

Geographers, it was argued in the introductory chapter, have expressed more of an interest in the applied side of optimisation theory. Their role should therefore be predominantly to focus more explicitly on the application of optimisation theory to real world problems that include spatial considerations, and to evaluate critically the validity of assumptions made. Does this imply that geographers must confine themselves to borrowing optimisation research tools developed by other disciplines? On the contrary, research tools previously developed are often far too theoretical and rigid in nature to be readily applicable. Geographers need therefore to build on existing knowledge by developing their own optimisation research tools that are more flexible, and that can easily be applied to real world spatial problems.

Goodchild (1985: 12) notes that

> the pragmatic and real division between disciplines results more from a natural taxonomy of techniques for analysis and problem solving than from natural breaks in subject matter.

If this is true, then it is time that geographers stop borrowing research tools from other disciplines; instead of letting other disciplines develop spatial tools for them. Rather, it is time that geographers start developing their own tools for spatial analysis. Such tools could include optimisation techniques realisticly and readily applicable to real world problems.

Part of the objective of this thesis has been to suggest a number of ways in which geographers can increase the flexibility and adaptability of spatial optimisation theory. One approach that has been stressed is that of utilising a multiobjective approach. The general advantages to be derived when utilising a multiobjective approach were outlined in Chapter Two. Briefly re-emphasising the main point, a multiobjective approach allows optimisation problems to be defined less narrowly and rigidly, and it more closely resembles decision making processes by placing the emphasis not on one unambiguous optimal answer, but on trade off and conflict resolution.

Some of the groundwork to multiobjective spatial analysis has been laid by a number of faculty members and past doctoral students at the Department of Geography and Environmental Engineering, the Johns Hopkins University, Baltimore (notably J. Cohon, J. Current, D. Schilling and C. ReVelle). It has been part of the objective of this thesis to stress some of this groundwork, and to further develop this field of research by discussing a multiobjective approach to solving travelling salesman type routing problems.

The scope for future theoretical research in multiobjective spatial optimisation analysis is broad. At the specific level, a number of future research projects can be developed from the MVP problem researched in this thesis. The present MVP problem definition, for example, assumes that if a node is visited, then all of that node's reward potential must be collected. An alternative problem definition would be to relax this assumption, allowing the collection of partial reward. Other examples of future research possibilities include an enquiry into multiobjective chromatic travelling salesman problems, and multiobjective travelling salesman problems that involve the search for one or a number of optimal depot locations. At a broader level, scope for future multiobjective spatial optimisation analysis includes theoretical work on numerous other routing problems, and a multiobjective approach to location-allocation analysis. The applicability of a multiobjective approach to solving real world spatial problems, too, are numerous, offering ample opportunities in applied geography.

Goodchild (1985:13) suggests that the secure future of geography lies in the development and application of a useful set of tools and theories for the solution of spatial problems. Multiobjective optimisation analysis offers one such addition to the geographers' bag of tools, an addition that will greatly assist the discipline in becoming involved in applied work. It is hypothesised that geographers will show an increasing interest in the wealth of potential theoretical and applied research possibilities offered when adapting a multiobjective approach to spatial analysis.

8.2 Time Dependent Spatial Optimisation

Geographers, as has been noted in section 1.3, are placing a growing emphasis on the role of time in spatial analysis. The explicit incorporation of time in spatial optimisation theory has however received little attention to date, perhaps with the exception of some research into routing and scheduling problems mentioned in the introductory chapter. A possible reason for this lack of research is that the explicit inclusion of time usually considerably increases the mathematical and computational complexities of solution procedures. These types of problems do not readily submit to exact solution approaches, and they have therefore been somewhat neglected. Solution approaches based on heuristics, and on a man-machine interactive programming approach, are however quite suitable for solving these types of problems.

The TDVP problem researched in this thesis represents an example of a spatial optimisation problem that considers time explicitly. It represents another example of an area of research that geographers can develop to make optimisation theory more readily applicable to real world problems, and that will increase their set of spatial tools.

The TDVP problem, as defined in this thesis, contains many assumptions. A possible relaxation of one or a number of these assumptions offers ample scope for future research. One assumption is that, while the reward potential of a node is time dependent, this dependency only matters for the time of arrival. Reward potential is assumed to be uniform throughout the vendor's stay. This linear assumption is often unrealistic. Numerous research problems exist where demand may grow or decline during the vendor's stay. An analysis of problems where demand is nonlinear throughout the vendor's stay therefore offers one future avenue of research.

A second assumption is that the vendor cannot depart from a node unless all reward at that node has been collected. It may however be in the vendor's interest to depart from a node early to ensure arrival at some other node at peak demand.

The TDVP problem definition assumes that the reward potential of a node is depleted once it has been visited, and that every node ought therefore to be visited only once. What if reward is allowed to replenish? Given this possibility, individual nodes may be visited more than once, and an entirely new set of research questions can be posed.

What if the assumption that the vendor must always either travel, or service the nodes, is dropped? It may prove worthwhile to wait in order to arrive at a node at a specific time. Should reward be replenishable, then it may even prove worthwhile to remain at the same node, and to wait for reward to re-accumulate. This is the point at which the vendor will cease to be mobile, but will open a fixed store.

Finally, it is presently assumed that the same time dependent demand function applies to the entire problem space. What if each node contains a unique time-dependent demand function?

It becomes obvious from the above discussion that even a simple extension of the TDVP problem provides considerable scope for future research. Many more research possibilities can be found by extending the explicit inclusion of arrival times in the objective functions to other routing problems, and to spatial optimisation problems in general.

8.3 Conclusion

This thesis has identified a number of areas of enquiry thought to be of interest to geographers, and that appear to offer considerable research opportunities. These areas include a multiobjective research approach, the explicit inclusion of time and an interactive programming approach. The two problems addressed in this thesis, the MVP and TDVP problems, are argued to comprise part of these suggested research directions. Techniques suggested to solve these two problems utilised interactive programming. The advantages to interactive programming are obvious, and it is an approach that should be used more frequently in the future. The advantage to heuristics is that they allow MVP and TDVP problems to be solved that have to evaluate amongst a large set of nodes. Exact solution techniques presently do not allow this because of computer capacity limitations. Ongoing research in computer development, as for example the study of parallel processors (The Economist, 1985), may allow even large problems to be solved by exact methods in the future, for example by explicit enumeration. In the meantime, heuristics do however pose the most feasible solution approach.

It is the author's belief that the type of research outlined above, together with the development of other multiobjective and time specific spatial optimisation analysis, offers ample scope for future research within theoretical and applied geography, and that it will help to emphasize the importance of geographical research in optimisation theory.

BIBLIOGRAPHY

Abler, R., J.S. Adams and P. Gould, 1971. Spatial Organization, Prentice Hall, Englewood Cliffs, New Jersey.

Aneja, Y. and K. Nairn, 1979. "Bicriteria Transportation Problems," Management Science, 25, pp. 73-78.

Assad, A., M. Ball, L. Bodin and B. Golden, 1981. Combined Distribution Routing and Scheduling in a Large Commercial Firm, Proceedings of the 1981 Northeast AIDS Conference, (Edited by R. Pavan and P. Anderson), Boston, pp. 99-102.

Babin, A., M. Florian, L. James Lefebvre and H. Spiess, 1982. An Interactive Graphic method for Road and Transit Planning, Publication No.204, Centre de recherche sur les transports Department d'informatique et de recherche operationelle, Universite de Montreal.

Baker, E., 1980. Oriented Vehicle Routing and the Travelling Salesman Problem, School of Business, University of Miami, Florida.

Barber, G., 1976. "Land-Use Plan Design via Interactive Multi-Objective Programming," Environment and Planning, 8, pp. 625-633.

Barber, G., 1977. "Urban Population Distribution Planning," Annals of the Association of American Geographers, 67, pp. 239-245.

Beardwood, J., J.H. Halton and J.M. Hammersley, 1959. "The Shortest Path Through Many Points," Proceedings of the Cambridge Philosophical Society, 55, pp. 299-327.

Belenson, S. and K. Kapur, 1973. "An Algorithm for Solving Multicriterion Linear Programming Problems with Examples," Operational Research Quarterly, 24, pp. 65-77.

Bellmore, M. and S. Hong, 1974. "Transformation of Multi-Salesman Problem to the Standard Travelling Salesman Problem," Journal of ACM, 21, pp. 500-504.

Bellmore, M. and J. Malone, 1971. "Pathology of Travelling Salesman Subtour-Elimination Algorithms," Operations Research, 19, pp. 278-307.

Bellmore, M. and G.L. Nemhauser, 1968. "The Travelling Salesman Problem," Operations Research, 16, pp. 538-558.

Benayoun, R., J. deMontgolfier, J. Tergny and O. Laritchev, 1971. "Linear Programming with Multiple Objective Functions: Step Method," Mathematical Programming, 1, 366.

Bodin, L., 1975. "A Taxonomic Structure for Vehicle Routing and Scheduling Problems," Computers and Urban Society, 1, pp. 11-29.

Bodin, L., B. Golden, A. Assad and M. Ball, 1983. "Routing and Scheduling of Vehicles and Crews: The State of the Art," Computers and Operations Research, Special Issue, 10, No.2, pp. 63-211.

Boyde, Y., 1965. "Routing Methods: Principles for Handling Multiple Travelling Salesman Problems," Lund Studies in Geography, Series C, 5, Gleerup, Lund.

Briskin, L., 1966. A Method of Unifying Multilple-Objective Functions. Management Science, 12, pp. 406-416.

Bromley, R.J., R. Symanski and C.M. Good, 1975. "The Rationale of Periodic Markets," Annals of the Association of American Geographers, 65, pp. 530-531.

Carlstein, T., 1974. Time Allocation, University of Lund, Department of Geography, (mimeo).

Carlstein, T., 1975. Time Allocation: on the Capacity for Human Interaction in Space and Time, Department of Geography, Lund, (mimeo).

Carlstern, T., D.N. Parkes and N.J. Thrift (Eds.), 1978. Timing Space and Spacing Time, Vol. III, Time and Regional Dynamics, Edward Arnold, London.

Charnes, A. and W.W. Cooper, 1961. Management Models and Industrial Applications of Linear Programming, Wiley, New York.

Christofides, N., A. Mingozzi and P. Toth, 1981. "State Space Relaxation Procedures for the Computation of Bounds to Routing Problems," Networks, 11(2), pp. 145-164.

Church, R.L. and D.L. Huber, 1980. "Modelling to Generate Alternative Configurations to Multiobjective Public Facility Location Problems," paper presented at the 1980 North American Meetings of the Regional Science Association, Milwaukee, Wisconsin.

Clarke, G. and J.W. Wright, 1964. "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points," Operations Research, 12, pp. 567-581.

Cochrane, J. and M. Zeleny, 1973. Multiple Criteria Decision Making, 816 pp., University of South Carolina Press, Columbia.

Cohon, J.L., 1978. Multiobjective Programming and Planning, Mathematics in Science and Engineering, Vol. 140, Academic Press.

Cohon, J.L. and D. Marks, 1975. "A Review and Evaluation of Multiobjective Programming Techniques," Water Resources Res., 11(2), pp. 208-220.

Cooper, L., 1964. "Heuristic Methods for Location-Allocation Problems," SIAM Review, 6, pp. 37-53.

Cooper, L., 1968. "An Extension of the Generalised Weber Problem," Journal of Regional Science, 8, pp. 181-197.

Courtney, J.F. Jr., T.D. Klastorin and T.W. Ruefli, 1972. "A Goal Programming Approach to Urban-Suburban Location Preference," Management Science, 18, pp. 258-268.

Crowder, H. and M. Padberg, 1980. "Solving Large Scale Symmetric Travelling Salesman Problems to Optimality," Management Science, 26, pp. 495-509.

Current, J.R., 1981. Multiobjective Design of Transportation Networks, Ph.D. Dissertation (unpublished), Johns Hopkins University, Baltimore, Maryland.

Current, J., C. ReVelle and J. Cohon, 1982. "Multiobjective Design of Transportation Networks," Operations Research Group Report Series Report No. 82-02, Department of Geography and Environmental Engineering, Johns Hopkins University, Baltimore, Maryland.

Current, J., C. ReVelle and J. Cohon, 1983. "The Maximum Covering/Shortest Path Problem: A Multiobjective Network Design and Routing Formulation," Operations Research Group Report Series Report No. 83-03, Johns Hopkins University, Baltimore, Maryland.

Current, J., C. ReVelle and J. Cohon, 1985. "The Application of Location Models to the Multiobjective Design of Transportation Networks," Working Paper Series No. WPS 85-38, College of Administrative Science, The Ohio State University.

Dane, C.W., N.C. Meador and J.B. White, 1977. "Goal Programming in Land Use Planning," Journal of Forestry, 75, pp.325-329.

Dantzig, G.B., D.R. Fulkerson and S.M. Johnson, 1954. "Solution of a Large Scale Travelling Salesman Problem," Operations Research, 2, pp. 393-410.

Dantzig, G.B., D.R. Fulkerson and S.M. Johnson, 1959. "On a Linear Programming Combinatorial Approach to the Travelling Salesman Problem," Operations Research, 7, pp. 58-66.

Dantzig, G.B. and J.H. Ramser, 1959. "The Truck Dispatching Problem," Management Science, 6, pp. 80-91.

deNeufville, R. and R. Keeney, 1973. "Multiattribute Preference Analysis for Transportation Systems Evaluation," Transportation Research, 7, pp. 63-76.

Deutsche Zentrale fur Tourismus, Beethovenstrasse 69, D-6000, Frankfurt/Main 1, Auschnitt aus der Strassenkarte RV 90 Deutschland 1:800,000.

Dyer, J., 1972. "Interactive Goal Programming," Managemant Science, 19, pp. 62-70.

Dyer, J., 1973. "An Empirical Investigation of a Man-Machine Interactive Approach to the Solution of the Multiple Criteria Problem", In: Multiple Criteria Decision Making, (J. Cochrane and M. Zeleny, eds.), pp. 202-216. University of South Carolina Press, Columbia.

Ecker, J. and I. Kouada, 1975. "Finding Efficient Points for Linear Multiple Objective Programs," Mathematical Programming, 8, pp. 375-382.

Eilon, S., C.D.T. Watson-Gandy and N. Christofides, 1971. Distribution Management: Mathematical Modelling and Practical Analysis, Hafner, New York.

Elmaghraby, S., 1970. Some Network Models in Operations Research, Springer Verlag, New York.

Evans, J. and R. Steuer, 1973. "A Revised Simplex Method for Linear Multiple Objective Programs," Mathemathical Programming, 5, pp. 54-61.

Fandel, G. and T. Gal, 1980. Multiple Criteria Decision Making Theory and Application; Hagen, Springer-Verlag, Berlin and New York.

Farquhar, P., 1977. "A Survey of Multiattribute Utility Theory," In, Multiple Criteria Decision Making, (M. Starr and M. Zeleny, eds.), pp. 59-90, North-Holland, Amsterdam.

Garey, M.R. and D.S. Johnson, 1979. Computers and Intractability: A Guide to the Theory of NP Completeness, W.N. Freeman and Company, San Francisco.

Geoffrion, A., 1967. "Solving Bicriterion Mathematical Problems," Operations Research, 15, pp. 39-54.

Geoffrion, A., 1975. "A Guide to Computer-Assisted Methods for Distribution Planning," Sloan Management Review, 16(2), pp. 17-41.

Ghosh, A., 1980. "A Model of Periodic Marketing and Itinerant Trading," Paper presented at the Twenty-Seventh North American Meeting, Regional Science Association, Milwaukee, WI, November 14-16, 1980.

Ghosh, A., 1982. "A Model of Periodic Marketing", Geographical Analysis, 14, No. 2, pp. 155-166.

Gillett, B. and J. Johnson, 1974. "Sweep Algorithm for the Multiple Depot Vehicle Dispatch Problem," quoted in, R.C. Larson and A.R. Odoni, 1981, Urban Operations Research, Prentice Hall, Englewood Cliffs, New Jersey.

Golden, B.L., 1976. Large Scale Vehicle Routing and Related Combinatorial Problems, Ph.D. Dissertation (unpublished), MIT Operations Research Center, Cambridge, Mass..

Golden, B.L. and T. Magnanti, 1977. "Deterministic Network Optimisation: A Bibliography," Networks, 7, pp. 149-183.

Golden, B.L., L. Bodin, T. Doyle and W. Stewart, Jr., 1980. "Approximate Travelling Salesman Algorithms," Operations Research, 28, No. 3, Part II, pp. 694-711.

Gomory, R.E., 1958. "Outline of an Algorithm for Integer Solutions to Linear Programming," Bulletin of the American Mathematical Society, 64, pp. 275-278.

Gonzales, R.H., 1962. Solution to the Travelling Salesman Problem by Dynamic Programming on the Hypercube, Technical Report No. 18, Operations Research Center, Cambridge, Mass..

Goodchild, M.F., 1985. "Questions, Tools or Paradigms: Scientific Geography in the 1980's", Ontario Geography, 25, pp. 3-14.

Gupta, J., 1978. "A Search Algorithm for the Travelling Salesman Problem," Computers and Operations Research, 5, pp. 243.

Gupta, S.M. and J.M. Cozzolino, 1974. Fundamentals of Operations Research for Management, Holden-Day, San Francisco.

Hagerstrand, T., 1970. "What About People in Regional Science," Papers of the Regional Science Association, 24, pp. 7-21.

Hagerstrand, T., 1973. "The Domain of Human Geography," In: Chorley, R.J. (ed.), Directions in Geography, Methuen, London, pp. 67-87.

Hagerstrand, T., 1975. "Space, Time and Human Conditions," In: Karlquist, A., L. Lundquist and F. Snickars (eds), Dynamic Allocation of Urban Space, Farnborough, Saxon House, pp. 3-12.

Haimes, Y., 1973. "Integrated Systems Identification and Optimisation," In: Control and Dynamic Systems: Advances in Theory and Application, Academic Press, New York, Volume 9, pp. 435-518.

Haimes, Y., 1977. Hierarchial Analysis of Water Resource Systems, McGraw-Hill, New York.

Haimes, Y. and W. Hall, 1974. "Multiobjectives in Water Resource Systems Analysis: The Surrogate Worth Trade-Off Method", Water Resources, 10, pp. 615-624.

Haimes, Y., W. Hall and H. Freedman, 1975. Multiobjective Optimization in Water Resources Systems: The Surrogate Worth Trade-Off Method, Elsevier, Amsterdam.

Hansen, P., 1983. Essays and Surveys on Multiple Criteria Decision Making, Mons, Springer-Verlag, Berlin and New York.

Hartley, R.V., 1976. Operations Research: A Managerial Approach, Goodyear, Pacific Palisades, California.

Harvey, D., 1973. Explanation in Geography, Edward Arnold Limited, London.

Harvey, M.E., R.T. Hocking and J.R. Brown, 1974. "The Chromatic Travelling Salesman Problem and its Application to Planning and Structuring Geographic Space," Geographical Analysis, 6, pp. 33-52.

Hay, A.M., 1971. "Notes on the Economic Basis for Periodic Markets in Developing Countries," Geographical Analysis, 3, pp. 393-401.

Held, M. and R.M. Karp, 1971. "The Travelling Salesman Problem and Minimum Spanning Trees: Part II," Mathematical Programming, 1(1), pp. 6-25.

Hill, M., 1973. Planning for Multiple Objectives: An Approach to the Evaluation of Transportation Plans, Monograph No. 5, Regional Science Research Institute, Philadelphia, Pennsylvania.

Hocking, R.T., 1972. The Chromatic Travelling Salesman Problem, D.B.A. Dissertation, Kent State University.

Holl, S., 1973. Efficient Solutions to a Multicriteria Linear Program, with application to an Institution of Higher Education, 194 pp., Ph.D. Dissertation, Department of Mathematical Sciences, John Hopkins University, Baltimore, Maryland.

Janelle, D.G., 1968. "Central Place Development in a Time-Space Framework," Professional Geographer, 20, pp. 5-10.

Janelle, D.G., 1969. "Spatial Reorganisation: A Model and Concept," Annals of the Association of American Geographers, 58, pp. 348-364.

Janelle, D.G., 1976. "Stagecoach Operations in Maine, 1826 - 1829," Proceedings, New England - St. Lawrence Valley Geographical Society, 6, pp. 15-48.

Jara-Diaz, S. and A. Han, 1980. "Adapting Multiobjective Optimisation Methods to Transport Project Evaluation: A Departure from Benefit-Cost Analysis in Network Improvements," quoted in J.R. Current, 1981, Multiobjective Design of Transportation Networks, Ph.D Thesis, Johns Hopkins University, Baltimore, Maryland.

Jonnson, E., 1968. Studies in Multiobjective Decision Models, Studentlitteratur, Lund.

Johnston, R.J., 1979. Geography and Geographers, Edward Arnold Limited, London.

Karg, L.L. and G.L. Thompson, 1964. "A Heuristic Approach to Solving Travelling Salesman Problem," Management Science, 10, pp. 225-248.

Karp, R.M., 1977. "Probabilistic Analysis of Partitioning Algorithms for the Travelling Salesman Problem," Mathemathics and Operations Research, 2, pp. 209-224.

Keller, C.P., 1985. Routing to Cover Time Dependent Demand: A Multiobjective Approach, in Proceedings of the Sixteenth Annual Pittsburgh Conference on Modeling and Simulation, Pittsburgh, Pennsylvania.

Keeney, R. and H. Raiffa, 1976. Decisions with Multiple Objectives: Preference and Value Tradeoffs, 569 pp., Wiley, New York.

Killen, J.E., 1979. Linear Programming: The Simplex Method with Geographical Applications, Concepts and Techniques in Modern Geography Monograph Series Number 24, Geo Abstracts, Norwich.

Killen, J.E., 1983. Mathematical Programming for Geographers and Planners, St. Martins Press, New York.

Knuth, D.E., 1976. "Mathemathics and Computer Science: Coping with Finiteness," Science, 194(4271), pp. 1235-1242.

Koopmans, T.C., 1951. Activity Analysis of Production and Allocation, 404 pp., Wiley, New York.

Kornbluth, J., 1973. "A Survey of Goal Programming," Omega, 1, pp. 193-205.

Kornhauser, A.L. and J.L. Hess, 1978. "Interactive Graphics Sketch Planning Model for Urban Transportation," Transportation Research Record, p. 657.

Krolak, P., W. Felts and J. Nelson, 1972. "A Man-Machine Approach Towards Solving the Generalised Truck-Dispatching Problem," Transportation Research, 6, pp. 149-170.

Kruskal, J.Jr., 1956. "On the Shortest Spanning Subtree of the Graph and the Travelling Salesman Problem," Proceedings of the American Mathematical Society, 7, pp. 48-50.

Kuhn, H. and A. Tucker, 1951. Nonlinear Programming, Proceedings Berkeley Symposium on Mathematical and Statistical Probability, 2nd, (J. Neyman, ed.), University of California Press, Berkeley, pp. 481-492.

Kwak, N.K. and M.J. Schniederjans, 1979. "A Goal Programming Model for Improved Transportation Problem Solutions," Omega, 7, pp. 367-370.

Larson, R.C. and A.R. Odoni, 1981. Urban Operations Research, Prentice Hall, Englewood Cliffs, New Jersey.

Lee, S.M., 1972. Goal Programming for Decision Analysis, Auerbach, Philadelphia.

Lee, S. and L. Franz, 1979. "Optimising the Location-Allocation Problem with Multiple Objectives," International Journal of Physical Distribution and Mathematical Management, 9, pp. 245-255.

Lee, S.M., G. Green and C. Kim, 1981. "A Multiple Criteria Model for the Location-Allocation Problem," Computers and Operations Research, 8, pp. 1-8.

Lee, S.M. and L. Moore, 1977. "Multi Criteria School Bussing Models," Management Science, 23, pp. 703-716.

Leinbach, T.R. and R.G. Cromley, 1983. "A Goal Programming Approach to Public Investment Decisions: A Case Study of Rural Roads in Indonesia," Social and Economic Planning Science, 17, No. 1, pp. 1-10.

Lenntorp, B., 1976. "A Time-Space Structure Study of the Travel Possibilities of the Public Transport Passenger," Rapporter och Notiser, 24, Department of Geography, University of Lund.

Lin, S. and B. Kernigham, 1973. "An Effective Heuristic Algorithm for the Travelling Salesman Problem," Operations Research, 21, pp. 498-516.

Litke, J.D., 1984. "An Improved Solution to the Travelling Salesman Problem with Thousands of Nodes," Communications of the A.C.M., 27, No. 12, pp. 1227-1235.

Major, D., 1969. "Benefit-Cost Ratios for Projects in Multiple Objective Investment Programs," Water Resources Research, 5, pp. 1174-1186.

Marglin, S., 1967. Public Investment Criteria, MIT, Cambridge, Mass., 103 pp.

McGrew J.C. Jr., 1975. "Goal Programming and Complex Problem Solving in Geography," Papers in Geography, 12, Pennsylvania state University, University Park, Pennsylvania.

Monarchi, D., C. Kisiel and L. Duckstein, 1973. "Interactive Multiobjective Programming in Water Resources: A Case Study," Water Resources, 9(4), pp. 837-850.

Moore, L., B.W. Taylor and S. Lee, 1978. "Analysis of a Transshipment Problem with Multiple Conflicting Objectives," AIIE Transactions, 5, p. 333.

Morse, J.N., 1981. Organizations: Multiple Agents with Multiple Criteria, Nemark, Springer-Verlag, Berlin and New York.

Nijkamp, P., 1975. "A Multiobjective Analysis for Project Evaluation: Economic-Ecological Evaluation of a Land Reclamation Project," Papers of the Regional Science Association, 35, pp. 87-111.

Nijkamp, P., 1977. Theory and Application of Environmental Economics, North-Holland Publishing Company, Amsterdam.

Nijkamp, P., 1978. "Competition Among Regions and Environmental Quality," In: W. Buhr and P. Friedrich (eds.) Competition among Small Regions, Nomos Verlag, Baden-Baden, pp. 153-170.

Nijkamp, P., 1979. Multidimensional Spatial Data and Decision Analysis, Wiley, New York.

Nijkamp, P. and P. Rietveld, 1976. "Multiobjective Programming Models, New Ways in Regional Decision Making," Regional Science and Urban Economics, 6, pp. 253-274.

Nijkamp, P. and P. Rietveld, 1977. Impact Analyses, Spatial Externalities and Policy Choices, Research Memorandum 65, Department of Economics, Free University, Amsterdam.

Nijkamp, P. and P. Rietveld, 1979. Multilevel Multiobjective Models in a Multiregional System, Research Memorandum 3, Department of Economics, Free University, Amsterdam.

Orloff, C., 1974. "Routing a Fleet of M Vehicles to/from a Central Facility," Networks, 4, pp. 147-162.

Orloff, C., 1976. "Route constrained fleet scheduling," Transportation Science, 10, pp. 149-168.

Papadimitriou, C.H., 1977. "The Euclidean Travelling Salesman problem is NP-Complete," Theoretical Computer Science, 4, pp. 237-244.

Parkes, D.N. and N.J. Thrift, 1980. Times, Spaces and Places, Wiley, New York.

Psaraftes, H.N., 1978. A Dynamic Programming Approach to the Dial-A-Ride Problem, Report R78-34, MIT, Department of Civil Engineering, Cambridge, Mass.

Philip, J., 1972. "Algorithms for the Vector Maximisation Problem," Mathematical Programming, 2, p. 207.

Rapp, H.M., S. Piguet and A. Robert-Grandpierre, 1978. "Interactive Graphic System For Transit Route Optimisation," Transportation Research Record, 5, pp.30-36.

Relph, E., 1976. Place and Placelessness, Pion, London.

ReVelle, C.S. and R.W. Swain, 1970. "Central Facility Location," Geographical Analysis, 2, pp. 30-42.

Rietveld, P., 1980. Multiple Objective Decision Methods and Regional Planning, (Studies in Regional Science and Urban Economics: volume 7), North Holland Publishing Company,

Salkin, H.M., 1975. Integer Programming, Addison-Wesley, Reading, Mass..

Savas, E.S., 1978. "On Equity in Providing Public Services," Management Sciences, 24, pp. 800-808.

Schilling, D., 1976. Multiobjective and Temporal Considerations in Public Facility Location, Ph.D. Thesis, Department of Environmetal Engineering, John Hopkins University, Baltimore, Mass..

Schilling, D., C. ReVelle, J. Cohon and D. Elzinga, 1980. "Some Models for Fire Protection Location Decisions," European Journal of Operational Research, 5, pp. 1-7.

Schnieder, J.B., 1974. "Interactive Graphics in Transportation Systems Planning," DOT-TST-74-10, Final Report on a Seminar held in Seattle, Washington, 99 pp.. PB 227-264, NTIS, Springfield, Va..

Schuler, A.T. and J.C. Meadows, 1975. "Planning Resource Use on National Forests to Achieve Multiobjectives," Journal of Environmental Management, 3, pp. 351-366.

Schuler, A.T., H.H. Webster and J.C. Meadows, 1977. "Goal Programming in Forest Management," Journal of Forestry, 75, pp. 320-324.

Scott, A.J., 1971. Combinatorial Programming, Spatial Analysis and Planning, Methuen, London.

Stein, D.M., 1978. "An Asymptotic, Probabalistic Analysis of a Routing Problem," Mathematics of Operations Research, 3(2), pp. 89-101.

Steenbrink, P., 1974. Optimisation of Transportation Networks, 325 pp., Wiley, Bristol, England.

Student, K., 1976. "Cost vs. Human Values in Plant Location," Business Horizon, 19(II), pp. 5-11.

Swersey, A. and W. Ballard, 1982. Scheduling School Buses, Yale School of Organization and Management Technical Report, Yale University.

The Economist, 1885. Building a Parallel Computer, pp. 83-84.

Tillman, F. and T. Cain, 1972. "An Upper Bound Algorithm for Single and Multiple Terminal Delivery Problems," Management Science, 18, pp. 664-682.

Tuan, Yi-Fu, 1977. Space and Place: The Perspective of Experience, Edward Arnold, London.

Tuan, Yi-Fu, 1978. "Space, Time, Place: A Humanistic Frame,' In: T. Carlstein, D.J. Parkes and N.J. Thrift (eds.), Volume I, Making Sense of Time, Edward Arnold, London.

Turner, W., P. Ghare and L. Fourds, 1974. "Transportation Routing Problems, A Survey," AIIE Transactions, 6, pp. 288-301.

Tyagi, M.S., 1968. "A Practical Method for Truck Dispatching Problems," Journal of the Operations Research Society of Japan, 10, pp. 76-92.

Ullman, E., 1974. "Space and/or Time: Opportunity for Substitution and Prediction," Transactions Institute of British Geographers, 63, pp. 125-139.

van Delft, A. and P. Nijkamp, 1976. "A Multiobjective Decision Model for Regional Development, Environmental Quality Control and Industrial Land Use," Papers of the Regional Science Association, 36, pp. 35-57.

Van Delft, A. and P. Nijkamp, 1977. "Multi-Criteria Analysis and Regional Decision Making," Studies in Applied Regional Science, 8, Martinus Nijhoff, Leiden.

Wagner, H.M., 1969. Principles of Operations Research with applications to Managerial Decisions, Prentice Hall, Englewood Cliffs, New Jersey.

Webber, M.J. and R. Symanski, 1973. "Periodic Markets: An Economic Location Analysis," Economic Geography, 49, pp. 213-227.

Wren, A. and A. Holliday, 1972. "Computer scheduling of vehicles from one or more depots to a number of delivery points," Operations Research Quarterly, 23, pp. 333-344.

Yu, P. and M. Zeleny, 1975. "The Set of all Nondominated Solutions in Linear Cases and a Multicriteria Simplex Method," Journal Mathematical Anal. Appl., 49(2), pp. 430-468.

Zadeh, L., 1963. "Optimality and Non-Scalar-Valued Performance Criteria," IEEE Trans. Automatic Control AC-8, 59 pp.

Zeleny, M., 1974. Linear Multiobjective Programming, 220 pp., Springer Verlag, Berlin and New York.

Zeleny, M., 1975a. "Multiple Criteria Decision Making Bibliography," In: Zeleny, M., 1976, Multiple Criteria Decision Making, Kyoto 1975, pp. 291-321. Springer-Verlag, Berlin and New York.

Zeleny, M., 1975b. "Multicriteria Simplex Method: A Fortran Routine," In: Zeleny, M., 1976, Multiple Criteria Decision Making, Kyoto 1975, pp. 323-345. Springer-Verlag, Berlin and New York.

Zeleny, M., 1976, Multiple Criteria Decision Making, Kyoto 1975, 345 pp., Springer-Verlag, Berlin and New York.

Zionts, S., 1978. Multiple Criteria Problem Solving, Buffalo, Springer-Verlag, Berlin and New York.

Zionts, S. and J. Wallenius, 1976. "An Interactive Programming Method for Solving the Multiple Criteria Problem." Management Science, 22, pp. 652-663.

END

12·03·86

FIN